Meta Clustering Learning for Large-scale Unsupervised Person Re-identification

Xin Jin∗
jinxin@eias.ac.cn
Eastern Institute for Advanced Study

Tianyu He
deepitimhe@gmail.com
Alibaba Group

Tongliang Liu
tongliang.liu@sydney.edu.au
The University of Sydney

Xin chao Wang
xinchao@nus.edu.sg
National University of Singapore

Jianqiang Huang
ejianqiang.jqh@gmail.com
Alibaba Group

Xian-Sheng Hua
huaxiansheng@gmail.com
Alibaba Group

Zhibo Chen
chenzhibo@ustc.edu.cn
University of Science and Technology of China

ABSTRACT
Unsupervised Person Re-identification (U-ReID) with pseudo labeling recently reaches a competitive performance compared to fully-supervised ReID methods based on modern clustering algorithms. However, such clustering-based scheme becomes computationally prohibitive for large-scale datasets, making it infeasible to be applied in real-world application. How to efficiently leverage endless unlabeled data with limited computing resources for better U-ReID is under-explored. In this paper, we make the first attempt to the large-scale U-ReID and propose a “small data for big task” paradigm dubbed Meta Clustering Learning (MCL). MCL only pseudo-labels a subset of the entire unlabeled data via clustering to save computing for the first-phase training. After that, the learned cluster centroids, termed as meta-prototypes in our MCL, are regarded as a proxy annotator to softly annotate the rest unlabeled data for further polishing the model. To alleviate the potential noisy labeling issue in the polishing phase, we enforce two well-designed loss constraints to promise intra-identity consistency and inter-identity strong correlation. For multiple widely-used U-ReID benchmarks, our method significantly saves computational cost while achieving a comparable or even better performance compared to prior works.

CCS CONCEPTS
• Information systems → Top-k retrieval in databases.

KEYWORDS
Clustering, Unsupervised Person Re-identification, Computational Cost Saving

∗Corresponding Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '22, October 10–14, 2022, Lisboa, Portugal.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9203-7/22/10...$15.00
https://doi.org/10.1145/3503161.3547900

1 INTRODUCTION
Ubiquitous cameras generate innumerable pedestrian data every day. Due to the growing demands on person re-identification (ReID) and its expensive labeling cost, unsupervised person ReID (U-ReID) [9, 12, 14, 22, 30, 31, 33, 40, 49, 63, 65, 67, 78] has attracted increasing attention recently.

There are mainly two categories in U-ReID. One is unsupervised domain adaptive (UDA) person ReID, which first pre-trains a model on the labeled source dataset, and then fine-tunes the model on the unlabeled target dataset to reduce domain gap [11, 35, 55, 58, 65, 75, 76]. Albeit effective, UDA ReID branch typically suffers from a complex adaptation process, and its success also relies on an assumption that the discrepancy between source and target domain is not significant. This motivates the exploration on the other branch, the clustering-based unsupervised ReID [9, 14, 18, 22, 36, 54]. As the “Previous work” shown in Figure 1, the works of domain adaptive (UDA) person ReID, which first pre-trains a model on the labeled source dataset, and then fine-tunes the model on the unlabeled target dataset to reduce domain gap [11, 35, 55, 58, 65, 75, 76]. Albeit effective, UDA ReID branch typically suffers from a complex adaptation process, and its success also relies on an assumption that the discrepancy between source and target domain is not significant. This motivates the exploration on the other branch, the clustering-based unsupervised ReID [9, 14, 18, 22, 36, 54]. As the “Previous work” shown in Figure 1, the works of
which is challenging but valuable and meaningful to bridge the previous works [9, 22, 30] will take a memory usage up to 22GB, Sec. 3.1, 3.2 for details). In the first phase, the features of the prototype optimization and prototype-referenced polishing (see learning while alleviating noisy label issue for ReID model.

To further leverage the rest unlabeled data, we take the learned prototypes from partial data as proxy annotator to pseudo-label them, and then polish model based on such pseudo labels with two well-designed losses (as a minor contribution) to promise intra-identity consistency and inter-identity strong correlation, which helps alleviate the noisy label issue.

• As the first attempt to handle the large-scale unsupervised ReID, extensive experiments on multiple benchmarks show that MCL could significantly save computational cost while achieving a state-of-the-art performance. In particular, MCL achieves ReID performance improvements of 4.8%, 2.9% in mAP on the large-scale MSMT17 [58], LaST [45], but saves 71.8%/87.9% memory costs and 73.7%/85.7% time costs compared to the baselines.

The second prototype-referenced polishing is based on the learned meta-prototypes in the previous phase, which are taken as a proxy annotator to mine the potential label information for the rest unlabeled data. For each unused person image, we get a soft real-valued label likelihood vector by comparing it with meta-prototypes reference. Based on such clustering-free pseudo label, we further polish model by mining the relative comparative characteristic in person images. The reason why we call this phase as “polishing” is because “polish” has the meaning of try to perfect one’s skill, like here we promote the discriminative feature learning for ReID model with the rest unlabeled data.

Another point should be noticed is that, the pseudo labeling itself, no matter of clustering-based or reference-based may generate wrong label predictions [20, 30, 54, 60]. As shown in Figure 1(b), the larger size of unlabeled dataset, the more possible of generating noisy labels. To alleviate it, we further leverage two loss constraints for label denoising in MCL. One loss enforces instance-level consistency to reduce intra-latent variance and the other constructs a soft-weighted triplet constraint to promise inter-identity correlation. In this way, MCL could better investigate the discriminative information of data even with noisy pseudo labels. We summarize our main contributions as follows:

• To our best knowledge, this paper is the first to achieve the unsupervised large-scale ReID training while considering the computational cost savings. A “small data for big task” paradigm dubbed Meta Clustering Learning (MCL) is proposed. MCL performs clustering-based ReID training on partial unlabeled data, saving computing resources.

• To further leverage the rest unlabeled data, we take the learned prototypes from partial data as proxy annotator to pseudo-label them, and then polish model based on such pseudo labels with two well-designed losses (as a minor contribution) to promise intra-identity consistency and inter-identity strong correlation, which helps alleviate the noisy label issue.

2 RELATED WORK

2.1 Unsupervised Person Re-identification

Unsupervised Domain Adaptive (UDA) ReID. This branch usually utilizes transfer learning, e.g., style translation [77], to reduce domain gap between source and target ReID scenarios for adaptation [8, 11, 38, 58, 69]. Their performance is typically inferior to the
clustering-based approaches, since there is still a gap between the style-translated images and the realistic person images [56, 62].

**Clustering-based Unsupervised ReID.** This branch typically trains ReID model directly on unlabeled dataset with a clustering-based pseudo label estimation [14, 18, 46, 62, 63, 68, 72]. This clustering labeling and training process are usually alternatively performed until the model is stable. In particular, Lin et al. [36] treat each individual sample as a cluster, and then gradually group similar samples into one cluster to generate pseudo labels. Jin et al. [30] introduce a global distance-distribution separation constraint to handle the sample-wise noisy label. SPCL [22] proposes a self-paced contrastive learning framework to gradually create more reliable clusters for ReID training while updating the hybrid memory containing both source and target domain features. Similarly, ClusterContrast [9] further stores feature vectors inside a cluster-level memory to alleviate the inconsistent clustering issue. Recently, Isobe et al. [28] introduce cluster-wise contrastive learning (CCL), progressive domain adaptation (PDA), Fourier augmentation (FA), and ICE [3] introduces inter-instance contrastive encoding to boost the existing class-level contrastive ReID methods. However, all these methods focus on how to get more reliable pseudo labels or how to better leverage them for discriminative feature learning, an important point of computational cost is still under-explored.

Besides, the noisy pseudo label issue in these methods has not yet been well addressed.

**Reference-based Pseudo Labeling in ReID.** Existing representative works, like MAR [65], MMCL [54], MPRD [29], and SSL [37] either use labeled source data for pseudo labels generation or assign each unlabeled person image with a multi-class/softened label via pairwise similarity computation. Differently, our MCL creates clustering-free soft pseudo labels with the reference of online updated meta-prototypes that stored in the memory. Such design is more efficient because it does not need source labeled dataset as reference, and meta-prototypes (like a FC layer) could help directly infer out real-valued labels instead of repeat pairwise comparisons. Moreover, more reliable meta-prototypes encourage more accurate pseudo labeling (more effective unsupervised training), and vice versa. They promote each other, achieving a win-win effect.

### 2.2 Self-supervised Representation Learning

MCL is also related to self-supervised representation learning (SSL). Based on contrastive learning framework, SSL has achieved a great success [15, 41], e.g., MoCo [24], MoCov2 [6], SimCLR[4], SimCLRv2 [5], BYOL [23], and SimSiam [7]. Their main idea is to match a same instance in different augmented views, which typically relies on a large number of explicit pairwise feature comparisons and faces a computational challenge. Besides, these instance-wise SSL methods cannot directly address the fine-grained unsupervised ReID problem (they can only be taken for pre-training/initiation [16, 17, 64]), because ReID needs the cluster priors to mine fine discriminative clues.

### 3 META CLUSTERING LEARNING (MCL)

**Overview.** To tackle the computing challenge in large-scale U-ReID, we propose a meta clustering learning (MCL), which is a unified episodic training framework, and comprises two phases of meta-prototype optimization (Figure 2) and prototype-referenced polishing (Figure 3). MCL alternates between these two phases: (1) group the partial unlabeled data into clusters and store the learned meta-prototypes, while training model with cluster-level contrastive loss (Section 3.1); (2) use meta-prototypes as reference to annotate the rest unlabeled samples for further fine-tune, and two loss constraints are enforced to promise intra-identity consistency and inter-identity correlation (Section 3.2).

Given an unlabeled dataset \( \mathcal{X} \), MCL first splits \( \mathcal{X} \) into \( N \) subsets uniformly, and then randomly selects one as meta-training subset \( \mathcal{X}_1 \) for meta-prototype optimization, the rest subsets \( \mathcal{X}_2, \mathcal{X}_3, \ldots, \mathcal{X}_N \) are taken for prototype-referenced polishing. This split is performed before each training epoch.

### 3.1 Phase 1: Meta-prototype Optimization

MCL costs less resources in clustering by only using a meta-training subset \( \mathcal{X}_1 \).

**Feature Extraction and Clustering.** As shown in Figure 2, a network \( f_0 \) (e.g., ResNet-50 [25], initialized with pre-trained weights on ImageNet [9, 21, 22, 28]) is taken as backbone to extract features from \( \mathcal{X}_1 \). Then, DBScan [13] is used to cluster these features (unclustered outliers are discarded [3, 9]). The IDs of cluster results are assigned to unlabeled samples as the pseudo labels for training.

**Query Setup and Meta-prototype Initialization.** After obtaining clustered pseudo labels for \( \mathcal{X}_1 \), we sample \( P \) person identities and \( I \) instances for each identity, to set up a mini batch with the size of \( P \times I \). Different from works [30, 46] that directly use the instance-wise loss contraints (e.g., triplet loss [27]) for training, we take each batch as a query set and employ a memory dictionary based contrastive learning [9, 22, 28] for optimization.

We maintain a group of learnable meta-prototypes \( \{\mathbf{w}_1, \ldots, \mathbf{w}_K\} \) stored in the memory dictionary. Here, \( K \) is same with the number of clustered clusters, which is always changing during the training. Particularly, the clustering algorithm (e.g., DBScan) is performed before each training epoch, and then the epoch-wise meta-prototypes are initialized with the mean feature vectors of each cluster, i.e., \( \mathbf{w}_k = \frac{1}{|B_k|} \sum v_i \), where \( v_i \) means \( i \)-th feature vector of \( k \)-th cluster, \( B_k \) denotes the \( k \)-th set that contains all the feature vectors within \( k \)-th cluster, and \( |\cdot| \) denotes the number of features in the set.

**Meta-prototypes Update and Model Optimization.** At each iteration \( t \) of epoch, the encoded feature vectors \( \{\mathbf{q}_i\} \) of \( P \times I \) query images in each mini-batch would be involved in meta-prototypes update. With the momentum updating [24], the \( k \)-th cluster prototype \( \mathbf{w}_k \) is updated by the mean of encoded query features belonging to class \( k \),

\[
\mathbf{w}_k^t \leftarrow m \cdot \mathbf{w}_k^{t-1} + (1 - m) \cdot \frac{1}{|B_k|} \sum_{\mathbf{q}_i \in B_k} \mathbf{q}_i^t,
\]

where \( B_k \) denotes the feature vector set belonging to class \( k \) in the mini-batch at the \( t \)-th iteration, and \( m \in [0, 1] \) is a momentum coefficient, which is empirically set as 0.2 following [9, 22]. The learned meta-prototypes are taken for model optimization together with query samples in this phase, and also play a role of proxy annotator (see the ‘robot’ in Figure 2,3) for the rest unlabeled subsets \( \mathcal{X}_2, \mathcal{X}_3, \ldots, \mathcal{X}_N \) in the next phase.
To achieve the savings of clustering cost, the first meta-prototype optimization phase only uses a part of unlabeled data, i.e., the meta-training subset $X_1$. The rest unlabeled subsets $X_2, X_3, \ldots, X_N$ are leveraged in a clustering-free manner in this second phase. Basically, we take the learned meta-prototypes $\{w_1, \ldots, w_K\}$ from the first phase as proxy annotator to softly mine discriminative information for those rest unlabeled data for model polishing. This two-phase training is equivalent to traverse the entire dataset once, i.e., one epoch.

Prototype-referenced Labeling. For clarity, we take an unused and unlabeled subset $X_2$ as example for illustration. Given $X_2 = \{x_i\}_{i=1}^{N_2}$, where each $x_i$ is a collected unlabeled person image, the learned meta-prototypes $\{w_1, \ldots, w_K\}$ defines the pseudo label space for $X_2$ as $[1, K]$. As shown in the ‘robot’ in Figure 3, the meta-prototypes set $\{w_k\}_{k=1}^{K}$ is taken as proxy annotator, a soft real-valued pseudo label $y^j_i$ can be assigned for $x_i$ by comparing $f(x_i)$ with the reference agents $\{w_k\}_{k=1}^{K}$. This soft prototype-referenced pseudo labeling process is,

$$y^j_i = L(f(x_i), \{w_k\}_{k=1}^{K})^j = \frac{\exp(w^T_j f(x_i))}{\sum_k \exp(w^T_k f(x_i))},$$

where $L(\cdot)$ means a soft pseudo labeling function. This function is epoch-wise and acts like a dimension-variable FC layer (i.e., dot-product), $y^j_i \in (0, 1)$ is the $j$-th entry of $y_i$. All dimensions of $y_i$ add up to 1 and each dimension represents the label likelihood.
is enforced to promise the correct relative correlation among person identities. Let \( \{x^d, x^p, x^n\} \) be an input triplet sample and the corresponding feature embeddings are \( \{f(x^d), f(x^p), f(x^n)\} \), the soft-weighted triplet loss is given by,

\[
L_{tri}^{sw} = \omega(a, p, n) [||f(x^d) - f(x^p)||_2^2 - ||f(x^d) - f(x^n)||_2^2 + m] +\
\omega(a, p, n) = \langle f(x^d), f(x^p) \rangle \langle f(x^d), f(x^n) \rangle
\]

where \( \omega(a, p, n) \) and \( m \) are the loss weighting factor and margin factor (0.3 by default), \( \langle \cdot, \cdot \rangle \) means the similarities between feature vectors, which adaptively alters the magnitude of the triplet loss in a soft manner. In general, when the anchor-positive pair is similar (i.e., \( \langle f(x^d), f(x^p) \rangle \) is high), the sample is more confident and reliable. Likewise, when the anchor-negative pair is similar (i.e., \( \langle f(x^d), f(x^n) \rangle \) is high), it forms a hard negative example [44]. Hence, \( L_{tri}^{sw} \) can give a higher priority and more attention on these reliable and hard cases, so as to alleviate the noisy label issue.

4 EXPERIMENT

4.1 Datasets and Implementation

Datasets and Evaluation. We evaluate the proposed MCL method on multiple ReID benchmarks (from small to large scale): PersonX (PX) [47], Market1501 (Ma) [70], MSMT17 (MT) [58], and the largest public ReID dataset (so far) LaST (LS) [45]. To further show the superiority of MCL in the large-scale data setting, we also conduct experiments on the mixed datasets, e.g., training on multiple datasets PX+Ma+MT+LS while testing on unseen test set of MT. The details about datasets are shown in Table 1.

Implementation Details. The proposed MCL is generic and can be applied to different clustering-based U-ReID backbones. Here, we re-implement ClusterContrast [9] as baseline, since it has been dominating the leaderboard in multiple benchmarks w.r.t unsupervised ReID performance, and is considerably more efficient as a source-free purely unsupervised ReID pipeline compared to those competitive adaptive (source data needed) U-ReID algorithms, like MMT [21], SpCL [22], [71] etc. ResNet-50 [25] is adopted as the backbone of the feature extractor and initialize the model with the parameters pre-trained on ImageNet [10].

At the beginning of MCL training, we first train the ReID model only with the first phase of meta-prototype optimization (skip the second phase of prototype-referenced polishing), which aims to warm up the meta-prototype learning, like the FC layer warm-up [25, 43], so as to have a reasonable pseudo labeling for the next model polishing. This process lasted for 5 epochs for PersonX, and 10 epochs for Market1501, MSMT17, and LaST. For image size, the input is resized as 256×128 (height×width) for all person datasets. For data augmentation, we perform random horizontal flipping, padding with 10 pixels, random cropping, and random erasing [74]. For batch size, each mini-batch contains 256 images of 16 pseudo person identities

### Table 1: Introduction and comparison of datasets we used.

| Dataset     | Style | Train IDs | Train images | Test IDs | Query images | Total images | Cameras |
|-------------|-------|-----------|--------------|----------|--------------|--------------|---------|
| PersonX [47]| Synthetic | 410 | 9,840 | 856 | 5,136 | 45,792 | 6 |
| Market-1501 [70]| Real | 751 | 12,958 | 750 | 3,368 | 32,688 | 6 |
| MSMT17 [58]| Real | 1,041 | 32,621 | 3,060 | 11,659 | 126,441 | 15 |
| LaST [45]| Real | 5,000 | 70,923 | 5,803 | 10,173 | 228,156 | 1 |
Table 2: Memory & Time Cost vs. Unsupervised ReID Performance (%). In which, M(MB), T(s) denotes the memory cost, time cost of performing clustering once in training, where ‘s’ means ‘second’. T(h) denotes the total training time where ‘h’ means ‘hour’.

We compare several MCL variants to baseline (All, i.e., Full Clustering scheme) by using 50%, 33%, 25%, and 20% data randomly selected from the entire unlabeled dataset as meta-training subset $X_1$. For the smallest dataset PersonX [47], it is not necessary to do experiments with too harsh computational requirements (e.g., 33%, 25%, 20%). We can see that the larger size of unlabeled dataset, the more superior of our method (red). Note that, the DukeMTMC-ReID dataset [42] has been taken down and thus not used in our experiment, we just use PersonX [47], Market1501 [70], MSMT17 [58], and LaST [45] for experiments.

| Methods | PersonX (9.8k imgs, 410 IDs) | Market1501 (12.9k imgs, 751 IDs) | MSMT17 (32.6k imgs, 1041 IDs) | LaST (71.2k imgs, 5000 IDs) |
|---------|-----------------------------|---------------------------------|-------------------------------|------------------------------|
|         | mAP | Rank1 | M (MB) | T (s) | T (h) | mAP | Rank1 | M (MB) | T (s) | T (h) | mAP | Rank1 | M (MB) | T (s) | T (h) | mAP | Rank1 | M (MB) | T (s) | T (h) |
| All     | 88.5 | 95.8 | 822.5 | 30.0 | 2.7  | 83.3 | 93.0 | 876.3 | 34.3 | 2.9  | 33.4 | 62.9 | 6251.5 | 118.3 | 9.3  | 19.8 | 74.0 | 22398.5 | 494.8 | 42.0 |
| 50%     | 79.0 | 93.5 | 412.6 | 13.1 | 2.2  | 82.9 | 92.7 | 348.6 | 10.8 | 2.4  | 38.2 | 66.5 | 1761.3 | 31.1 | 4.6  | 20.0 | 74.9 | 5779.6 | 121.2 | 20.0 |
| 33%     | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – |
| 25%     | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – |
| 20%     | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – | – – – – – – – – |

(16 instances for each person). During the training, we adopt Adam optimizer to train the ReID model with weight decay 5e-4. The initial learning rate is set to 3.5e-4, and is decayed by a factor of 0.1 every 20 epoch in a total of 60 epoch. Following [9, 22], we use DB-Scan and Jaccard distance [73] to cluster with $k$ nearest neighbors, where $k=30$. For DBScan, the maximum distance $d$ between two samples is experimentally set as 0.4 for market1501, 0.7 for other datasets, and the minimal number of neighbors in a core point is all set as 4.

4.2 Effectiveness and Necessity of MCL

**Memory & Time Cost vs. U-ReID Performance.** Table 2 shows the U-ReID performance resulted by using subsets of size 50%, 33%, 25%, and 20% randomly selected from all unlabeled data as meta-training set $X_1$ vs. directly conducting clustering over full data (All). We observe that using partial data for clustering with MCL effectively saves computational costs on both memory and time. For example, the MCL, 50% schemes nearly achieve the same ReID performance but save memory & time cost by over 50%, such savings are particularly obvious on the large datasets: 1761.3MB, 31.1s vs 6251.5MB, 118.3s on MSMT17 and 5779.6MB, 121.2s vs 22398.5MB, 494.8s on LaST. However, we also observe that the ReID performance of mAP/Rank1 has a noticeable drop, especially on the small dataset PersonX (‘blue’ in Table 2). We analyze that’s because on the noisy label issue will be enlarged on small datasets.

Interestingly, in contrast to the trend in Memory & Time saving vs. ReID accuracy reduction, we find an opposite trend for mAP/Rank1 improvements on the two largest datasets MSMT17 and LaST (‘red’ in Table 2). This reveals that the larger of unlabeled dataset, the more superior of our method. We analyze such gains come from two aspects: 1) less meta-training data gets more reliable clustered results; 2) the prototype-referenced polishing with intra- and inter-identity constraints promotes the discriminative ReID feature learning.

**Necessity of MCL.** Someone may think of directly splitting a large-scale dataset into multiple small subsets to do clustering-based U-ReID sequentially. This is also the most straightforward solution to handle the computational issue we focused. To study its feasibility, we deliberately design a scheme named *Naive Splitting Training*, where multiple subsets picked from one single large data set are sequentially used for clustering $\rightarrow$ labeling $\rightarrow$ training (same with phase-1 in MCL).

### Table 3: Study on the necessity of our MCL.

| Methods | Market1501 | MSMT17 |
|---------|------------|--------|
|         | mAP | Rank1 | mAP | Rank1 |
| 50%     | 73.6 | 82.2 | 22.2 | 43.9 |
| MCL     | 82.9 (↑16) | 92.7 (↑10) | 38.2 (↑38) | 66.5 (↑22) |
| 25%     | 68.4 | 74.1 | 16.2 | 36.5 |
| MCL     | 75.4 (↑16) | 89.3 (↑15) | 25.9 (↑12) | 53.4 (↑19) |

Training, where multiple subsets picked from one single large data set are sequentially used for clustering $\rightarrow$ labeling $\rightarrow$ training. *Naive Splitting Training* also could save memory cost due to its subset-wise clustering, but this operation also inadvertently enlarges the negative effect of time consuming and noisy labeling. As shown in Table 3, two *Naive Splitting Training* schemes of using 50%/25% subset as training unit, are inferior to MCL by 16.0%/9.7% in mAP on MSMT17, which reveals two facts that 1) naively splitting the holistic large-scale dataset for sequential training is not optimal, and 2) MCL is necessary and more superior.

4.3 Study on Mixed Large-scale Datasets

As discussed in Sec. 4.2 and Table 2, the larger size of unlabeled dataset, the more superior of MCL. To fully study this point, we further construct two mixed large-scale training datasets $PX+Ma+MT$ and $PX+Ma+MT+LS$, and evaluate models on the unseen test set of MSMT17 (MT). Note that, we originally planned to perform such group of experiments on the larger realistic ReID datasets, but which is limited by the truth that most large-scale realistic ReID datasets (e.g., Person30K [1], FastHuman [26]) have not fully released. As shown in Table 4, we can get two observations: 1) the scheme of All on the largest dataset $PX+Ma+MT+LS$ is failed to be directly clustered/trained due to the computing pressure. 2) MCL outperforms All by 2.3% in mAP under 50% on $PX+Ma+MT\rightarrow MT$ 3).

MCL performs better on $PX+Ma+MT$ than that on $PX+Ma+MT+LS$, which reveals two facts that 1) naively splitting the holistic large-scale dataset for sequential training is not optimal, and 2) MCL is necessary and more superior.
Table 4: Study on mixed large-scale datasets, where PX, Ma, MT, LS denotes PersonX, Market1501, MSMT17, LaST. Note that, in the experimental environment with four 16GB Tesla V100 GPUs, the scheme of All on PX+Ma+MT+LS is failed to be directly clustered/trained due to the computing pressure.

| TrainDatasets | Methods | Test: MT |
|---------------|---------|----------|
|               |         | mAP  | Rank1 | M (MB) | T (s) |
| MT            | All     | 33.4 | 62.9  | 6251.5 | 118.3 |
| PX+Ma+MT      | All     | 29.6 | 56.3  | 29788.7 | 244.8 |
|               | MCL 50% | 31.9 | 59.3  | 7698.3 | 82.6  |
|               | MCL 25% | 23.1 | 49.6  | 1399.0 | 35.9  |
| PX+Ma+MT+LS   | All     | –    | –     | –      | –     |
|               | MCL 50% | 25.5 | 49.9  | 21207.8 | 323.3 |
|               | MCL 25% | 17.1 | 39.5  | 5126.9 | 107.5 |

which may be due to the style/domain gap between LaST [45] and other ReID datasets.

4.4 Ablation Study

Influence of Loss Constraints. We study the effectiveness of the proposed siamese consistency loss $L_{sc}$ and soft-weighted triplet loss $L_{tri}$ in Table 5a. We see that MCL outperforms MCL w/o $L_{sc}$ by 4.0%/3.2% in mAP for 50%/25% settings on Market1501. When replace the soft-weighted triplet loss $L_{tri}$ with the basic triplet loss version [27], the scheme of MCL w/o $L_{tri}$ is inferior to MCL by 5.6%/4.4% in mAP for 50%/25% settings on Market1501. Such two constraints facilitate the pseudo label denoising via promising intra-identity consistency and inter-identity correlation. In addition, they are complementary and both vital to MCL, jointly resulting in a superior performance.

Influence of Data Split. As we described before Sec. 3.1, given an unlabeled dataset $X$, we use an random and uniform split strategy to divide the samples into meta training subset $X_1$ and the rest subsets $\{X_2, X_3, \ldots, X_N\}$. Such split is performed before each training epoch. And, the label spaces for different subsets $\{X_1, X_2, X_3, \ldots, X_N\}$ are same-size but non-overlapping. Here we study on the influence of different split designs. In Table 5b, the scheme of $MCL_{fixed}$ means we only conduct the data split once at beginning and fix the split results during the training. $MCL_{same}$ means the scheme where all subsets share the same label space. We can observe that $MCL_{fixed}$ is inferior to $MCL$ by 11.4%/37.6% in mAP under 50%/25% on Market1501, and $MCL_{same}$ is inferior to $MCL$ by 15.6%/11.5% in mAP under 25% on Market1501/MSMT17. We analyze that: 1) re-splitting dataset before each epoch plays a role of data re-organization, like the mechanism behind cross-validation [32], which avoids over-fitting and extremely cases, increasing robustness of MCL, 2) non-overlapped label spaces increase the diversity of training data, like a data augmentation, promoting the discriminative ReID representations learning. Such design brings obvious improvements especially when using less meta-training data. For example, MCL outperforms $MCL_{same}$ by only 2.1%/1.7% in mAP under 50%, but by 15.6%/11.5% under 25%. More analytic and ablated results (including limitation discussions) are presented in Supplementary.

Figure 4: Visual results of the same pseudo-labeled images on large-scale MSMT17 (two groups on the left) and LaST (two groups on the right), mined by using the general clustering technique and our meta-prototype referenced labeling. Green boxes denote correct results while red boxes denote false results. Important fine-grained feature clues are highlighted below each image pair. All faces in the images are masked for anonymization.

Figure 5: (a) Visualization of the percentages of correctly (red) and wrongly (green) pseudo labeling on MSMT17 as training goes on. (b) Visualization of t-SNE distributions on MSMT17. Different colors and digits represent different identities.

4.5 Visual Results and Insights

Visualization on Pseudo Labeling. To further show the proposed prototype-referenced labeling in MCL is superior to the general clustering, we compare these two pseudo-labeling methods by showing the same pseudo-labeled images (i.e., the positive pairs) in Figure 4. Our scheme is $MCL$, 50%. We observe that: 1) for the general clustering, the grouped entries share the global visual similar appearance. This is not reliable enough. For example, in the most left pair of Figure 4, the two women are dressed very similarly, the only local discriminative clue is they take different items in their hands. 2) the proposed meta-prototype referenced labeling has the capability of discovering fine-grained discriminative clues (bottom in Figure 4) due to the usage of relative comparative characteristic among samples. This also explains why MCL outperforms the baseline scheme of All even with less data for clustering to some extent.

Moreover, we also count the proportions of correctly and wrongly clustering persons into the same category on MSMT17 in Figure 5 (a), we can see that $MCL$, 50% could achieve a better identity grouping performance more quickly compared to the baseline scheme of All.

Visualization of Feature Distributions. In Figure 5 (b), we visualize the distributions of the features using t-SNE [50] on MSMT17. We compare the feature distribution with the baseline scheme of
Table 5: Ablation study for meta clustering learning (MCL).

(a) Study on the two loss constraints.

| Methods | Market1501 | MSMT17 |
|---------|------------|--------|
|         | mAP Rank1 | Rank5  | Rank10 |
| MCL w/o $L_{cc}$ | 78.9 | 88.8 | 35.1 | 62.8 |
| MCL w/o $L_{cc,m}$ | 77.3 | 86.4 | 33.4 | 60.7 |
| MCL      | 82.9 | 92.7 | 38.2 | 66.5 |

(b) Study on the data split strategies.

| Methods | Market1501 | MSMT17 |
|---------|------------|--------|
|         | mAP Rank1 | Rank5  | Rank10 |
| MCL fixed | 71.5 | 87.8 | 21.2 | 44.0 |
| MCL same | 80.8 | 92.0 | 36.5 | 63.8 |
| MCL      | 82.9 | 92.7 | 38.2 | 66.5 |

Table 6: Comparison with state-of-the-art methods on the unsupervised ReID, including purely unsupervised methods and unsupervised domain adaptation (UDA) methods. “None” represents the former, and other value represents the source-domain dataset in UDA method.

(a) Experiments on Market1501.

| Methods | Market1501 |
|---------|------------|
|         | source mAP Rank1 Rank5 Rank10 |
| BUC [36] | None 38.3 66.2 79.6 84.5 |
| UGA [59] | None 70.3 87.2 - - |
| SSL [37] | None 37.8 71.7 83.8 87.4 |
| MMCL [54] | None 45.5 80.3 89.4 92.3 |
| HCT [66] | None 56.4 80.0 91.6 95.2 |
| DG-Net [79] | MT 64.6 83.1 91.5 94.3 |
| CycAs [57] | None 64.8 84.8 - - |
| MMT [21] | MT 75.6 89.3 95.8 97.5 |
| SPCL [22] | None 73.1 88.1 95.1 97.0 |
| SPCL [22] | MT 77.5 89.7 96.1 97.6 |
| MPRD [29] | None 51.1 83.0 91.3 93.6 |
| ICE [3] | None 79.5 92.0 97.0 98.1 |
| HCD [71] | MT 80.2 91.4 - - |
| Cluster [9] | None 82.6 93.0 97.0 98.1 |
| MCL, 50% | None 82.9 92.7 97.6 98.7 |

(b) Experiments on MSMT17.

| Methods | MSMT17 |
|---------|--------|
|         | source mAP Rank1 Rank5 Rank10 |
| TAUDL [33] | None 12.5 28.4 - - |
| MMCL [54] | None 11.2 35.4 44.8 49.8 |
| UTAL [34] | None 13.1 31.4 - - |
| UGA [59] | None 21.7 49.5 - - |
| MMT [21] | Ma 24.0 50.1 63.5 69.3 |
| CycAs [57] | None 26.7 50.1 - - |
| SPCL [22] | None 19.1 42.3 55.6 61.2 |
| MPRD [29] | None 14.6 37.7 51.3 57.1 |
| HCD [71] | None 26.9 53.7 65.3 70.2 |
| ICE [3] | None 29.8 59.0 71.7 77.0 |
| Cluster [9] | None 33.3 63.3 73.7 77.8 |
| MCL, 50% | None 38.2 66.5 75.2 79.7 |

All, and observe that the features of different identities are better clearly separated for our scheme MCL, 50%, which demonstrates our learned ReID representations are more discriminative.

4.6 Comparison with State-of-the-arts

Although this work is the first attempt to achieve the unsupervised ReID learning while considering the computational cost savings, we also compare MCL to the state-of-the-art U-ReID methods that without considering resource limitations. From Table 6, we can see that MCL, 50% using only 50% unlabeled data for meta-clustering achieves a comparable U-ReID performance compared to SOTA methods, and even outperforms the second best ClusterContrast [9] by 4.9% in mAP on the large-scale MSMT17. In short, MCL is capable of achieving a good trade-off between U-ReID performance and computational costs.

Influence of Clustering Hyper-parameters. As discussed in implementation, we use DBScan and Jaccard distance [73] for first-phase training to cluster with $k$ nearest neighbors ($k=30$) following [9, 22]. For DBScan, the maximum distance $d$ between two samples is set as 0.4 for market1501, 0.7 for other datasets, and the minimal number of neighbors in a core point (denoted as $n$) is all set as 4. Here we analyze the influence of these parameters in Figure 7, and conclude that the proposed large-scale unsupervised ReID training method of MCL is robust and stable enough to achieve relatively satisfactory performance with variant hyper-parameters.

5 CONCLUSION

In this paper, we take the first attempt to explore a resource-friendly purely unsupervised person ReID framework, which effectively learns discriminative representations while considering the computational costs. A new concept of meta clustering learning (MCL) is introduced to perform clustering-based ReID training on partial unlabeled data, saving the required computing resources. For the rest data, we leverage the learned prototypes obtained before as proxy annotator to pseudo-label them. Based on the generated soft pseudo labels, we then polish model with two well-designed losses that take intra- and inter-identity constraints into account for alleviating noisy labels. MCL achieves a SOTA performance on unsupervised ReID, and could also flexibly meet the computing budgets in practice.

6 ACKNOWLEDGEMENTS

This work was supported in part by NSFC under Grant U1908209, 62021001, the National Key Research and Development Program of China 2018AAA0101400, and NUS Faculty Research Committee Grant (WBS:A-0009440-00-00).
[58] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. 2018. Person transfer GAN to bridge domain gap for person re-identification. In CVPR.

[59] Jutlin Wu, Yang Yang, Hao Liu, Shengcui Liao, Zhen Lei, and Stan Z Li. 2019. Unsupervised graph association for person re-identification. In ICCV. 8321–8330.

[60] Yiming Wu, Xiantian Wu, Xi Li, and Jian Tian. 2021. MGH: Metadata Guided Hypergraph Modeling for Unsupervised Person Re-identification. In ACMMM. 1571–1580.

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

[62] Fengxiang Yang, Ke Li, Zhun Zhong, Zhiming Luo, Xing Sun, Hao Cheng, Xiaowei Guo, Feiyue Huang, Rongrong Ji, and Shaozi Li. 2020. Asymmetric Co-Teaching for Unsupervised Cross Domain Person Re-Identification. In AAAI (2020).

[63] Qize Yang, Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. 2019. Patch-Based Discriminative Feature Learning for Unsupervised Person Re-Identification. In CVPR.

[64] Kaiwei Zeng, Munan Ning, Yaohua Wang, and Yang Guo. 2020. Hierarchical clustering with hard-batch triplet loss for person re-identification. In CVPR. 13657–13665.

[65] Yunpeng Zhai, Shijian Lu, Qixiang Ye, Xuebo Shan, Jie Chen, Rongrong Ji, and Yonghong Tian. 2019. Unsupervised Person Re-identification by Soft Multilabel Learning. In CVPR.

[66] Jun-Yan Zhu, Taesung Park, Phillip Isola, et al. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV. 2017–2024.

[67] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. 2021. Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In ACMMM. 433–441.

[68] Xinyu Zhang, Jiewei Cao, Chunhua Shen, and Mingyu You. 2019. Self-training with progressive augmentation for unsupervised cross-domain person re-identification. In ICCV.

[69] Xinyu Zhang, Dongdong Li, Zhigang Wang, Jian Wang, Errui Ding, Javen Qinfeng Shi, Zhaoliang Zhang, and Jingdong Wang. 2022. Implicit Sample Extension for Unsupervised Person Re-Identification. In CVPR. 7369–7378.

[70] Liang Zheng, Luyue Shen, et al. 2015. Scalable person re-identification: A benchmark. In ICCV.

[71] Yi Zheng, Shixiang Tang, Guodong Teng, Yixiao Ge, Kuijao Liu, Jing Qin, Dongjian Qi, and Despeng Chen. 2021. Online Pseudo Label Generation by Hierarchical Cluster Dynamics for Adaptive Person Re-Identification. In ICCV. 8371–8381.

[72] Yi Zheng, Yong Zhou, Jiaqi Zhao, Ying Chen, Rui Yao, Bing Liu, and Abdulmalek El Saddik. 2022. Clustering Matters: Sphere Feature for Fully Unsupervised Person Re-identification. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 18, 4 (2022), 1–18.

[73] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. 2017. Re-ranking person re-identification with k-reciprocal encoding. In CVPR.

[74] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. 13001–13008.

[75] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. 2018. Generalizing a person retrieval model hetero-and homogeneously. In ICCV.

[76] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. 2019. Invariance matters: Exemplar memory for domain adaptive person re-identification. In CVPR. 598–607.

[77] Jun-Yan Zhu, Taesung Park, Phillip Isola, et al. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV.

[78] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. 2021. Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In ACMMM. 433–441.

[79] Yang Zou, Xiaodong Yang, Zhiding Yu, BVK Vijaya Kumar, and Jan Kautz. 2020. Joint disentangling and adaptation for cross-domain person re-identification. In ECCV. Springer, 87–104.