Discrepant and Multi-instance Proxies for Unsupervised Person Re-identification

Chang Zou\textsuperscript{1}  Zeqi Chen\textsuperscript{2}  Zhichao Cui\textsuperscript{3}  Yuehu Liu\textsuperscript{2}\textsuperscript{*}  Chi Zhang\textsuperscript{2}

\textsuperscript{1}School of Software Engineering, Xi’an Jiaotong University
\textsuperscript{2}Institute of Artificial Intelligence and Robotics, Xi’an Jiaotong University
\textsuperscript{3}Chang’an University

Abstract

Most recent unsupervised person re-identification methods maintain a cluster uni-proxy for contrastive learning. However, due to the intra-class variance and inter-class similarity, the cluster uni-proxy is prone to be biased and confused with similar classes, resulting in the learned features lacking intra-class compactness and inter-class separation in the embedding space. To completely and accurately represent the information contained in a cluster and learn discriminative features, we propose to maintain discrepant cluster proxies and multi-instance proxies for a cluster. Each cluster proxy focuses on representing a part of the information, and several discrepant proxies collaborate to represent the entire cluster completely. As a complement to the overall representation, multi-instance proxies are used to accurately represent the fine-grained information contained in the instances of the cluster. Based on the proposed discrepant cluster proxies, we construct cluster contrastive loss to use the proxies as hard positive samples to pull instances of a cluster closer and reduce intra-class variance. Meanwhile, instance contrastive loss is constructed by global hard negative sample mining in multi-instance proxies to push away the truly indistinguishable classes. Extensive experiments on Market-1501 and MSMT17 demonstrate that the proposed method outperforms state-of-the-art approaches.

1. Introduction

Unsupervised person re-identification (Re-ID) aims to retrieve images of a particular person across camera views and scenes without annotations \cite{35, 48}. Most unsupervised methods adopt a two-step alternating training scheme: 1) generating pseudo labels by $k$-nearest neighbor search \cite{34, 42} or clustering \cite{15, 13, 27, 43, 8}; 2) training the model based on a uni-proxy (i.e., cluster centroid \cite{9} or learnable weight \cite{13}) of each cluster. However, due to the intra-class variance and inter-class similarity caused by the changeable human pose, illumination, and camera views \cite{54}, a uni-proxy is often biased and confused, failing to fully and accurately describe the information of a cluster. As a result, the features learned based on the uni-proxy are not compact and have unclear cluster boundaries in the embedding space, which in turn affects the quality of clustering. In order to learn discriminative features, CAP \cite{36} subdivides each cluster to obtain multiple camera-aware proxies, pulling an instance (i.e., sample) closer to all proxies in the cluster to alleviate intra-class variance. The later works ICE \cite{2} and PPLR \cite{7} adopt the same strategy. Although these methods improve the compactness of clusters, they depend on extra labels and ignore the intra-class variance caused by factors other than camera views. On the other hand, several works \cite{46, 14, 7} focus on reducing inter-class similarity to learn discriminative features. They consider performing batch hard negative sample mining \cite{20} to promote inter-class separation. However, as shown in Figure 1, due to the randomness of sampling, the negative samples selected for a query from the mini-batch may not be true hard negatives in the global embedding space, and therefore cannot enlarge the inter-class separation of actual indistinguishable classes.

To reduce intra-class variance without relying on additional annotations, we propose to use several discrepant cluster proxies to complementarily represent a cluster. Each

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{An illustration demonstrating that global hard negatives are more effective than batch hard in promoting inter-class separation. Different shapes represent different classes. (a) The batch hard negatives are the ones easy to distinguish. b) Our global hard negatives are the truly hardest and most informative samples of indistinguishable classes.}
\end{figure}
proxy concentrates on representing a portion of the information and the whole cluster is fully represented by several discrepant proxies. We obtain discrepant cluster proxies simply by updating the same cluster centroid with different update designs. Based on the cluster proxies, we propose cluster contrastive loss to increase the compactness of the clusters. As shown in Figure 2, \( \text{Proxy}_1 \) and \( \text{Proxy}_2 \) are the corresponding hard positive sample and easy positive sample for \( \text{Query}_1 \) according to pairwise similarity. Thus, contrastive loss enables \( \text{Proxy}_1 \) to generate a strong pull on \( \text{Query}_1 \) and \( \text{Proxy}_2 \) to generate a weak pull, resulting in \( \text{Query}_1 \) being closer to \( \text{Proxy}_1 \) after model optimization. Similarly, \( \text{Query}_2 \) will be closer to \( \text{Proxy}_2 \). As a result, \( \text{Query}_1 \) and \( \text{Query}_2 \) will become closer. Since the proxies are updated by these closer queries, \( \text{Proxy}_1 \) and \( \text{Proxy}_2 \) will also approach with training. Through the collaboration of two discrepant proxies, the cluster gradually gains intra-class compactness.

On the other hand, to further effectively decrease inter-class similarity while reducing intra-class variance, we propose to maintain finer-grained and more accurate multi-instance proxies through instance features of a cluster as the supplement of coarse-grained cluster proxies. Distinguished from the previous batch hard sample mining, the hard negative samples of a query are selected among the multi-instance proxies of all other classes with a global view. Then we exploit the true hard negatives to construct instance contrastive loss and purposefully increase the inter-class variance of indistinguishable classes.

Our contributions can be summarized as follows:
- We propose contrastive learning based on discrepant cluster proxies, which complementarily represent a cluster and collaboratively reduce intra-class variation.
- We propose global hard negative sample mining based on multi-instance proxies to select truly hard and informative negative samples to purposefully increase the inter-class variance of indistinguishable classes.
- Extensive experimental results with superior performance against the state-of-the-art methods demonstrate the effectiveness of the proposed method.

2. Related Work

Unsupervised Person Re-ID. Existing unsupervised methods can be roughly categorized into unsupervised domain adaptive (UDA) methods and purely unsupervised learning (USL) methods. UDA methods [13, 15, 14, 31, 44, 26, 35, 11, 52, 1, 21] transfer the knowledge learned from the labeled source domain to the unlabeled target domain. In contrast, USL methods [28, 34, 27, 43, 36, 41, 7, 46, 25, 45] is trained directly on unlabeled target datasets. Our method meets the more challenging USL setting. Recently, USL methods that generate pseudo labels by clustering and perform contrastive learning on cluster proxies have made great progress. SpCL [15] averages the instance features of a class in the memory bank as a uni-proxy for the class. Cluster-Contrast [9] directly stores a uni-proxy for each cluster to maintain the updating consistency. However, the cluster uni-proxy cannot effectively reduce the existing intra-class variance. Thus, CAP [36] forms multiple camera-aware proxies for each cluster to alleviate the camera domain gap. MCRN [39] stores multi-centroid representations for a cluster but only selects one as the proxy for a query to mitigate the effects of mixed clusters. Unlike these methods, we obtain several discrepant cluster proxies to completely represent a cluster and serve as hard positive samples to collaboratively enhance intra-class compactness.

Hard Sample Mining. Hard sample mining can improve training speed and performance [49]. Many recent unsupervised Re-ID methods utilize hard batch sample mining [20] to increase intra-class compactness and inter-class separation. MMT [14] and PPLR [7] learn hard samples by constructing softmax-triplet loss on the hardest positive and negative sample pairs. ICE [2] mines the hardest positive sample in the mini-batch and takes all samples of other identities as negatives to reduce intra-class variance. ISE [46] explores the hardest positive and negative samples among the original and generated samples within a batch. However, hard sample mining in the mini-batch does not consider global information of all classes. Therefore, we propose global hard negative sample mining based on multi-instance proxies to effectively enhance the inter-class variance among classes hard to discriminate.

Contrastive Learning. Contrastive learning [17, 6, 5, 32, 40, 16, 37] aims at maximizing the similarity of representations obtained from different distorted versions of a sample [16]. MoCo [17] builds a queued dictionary to keep an abundance of negative samples and introduces a momentum encoder to ensure their consistency. We perform both cluster-level and instance-level contrastive learning based on discrepant cluster proxies and multi-instance proxies. Like MoCo, We use a momentum encoder to keep the consistency of negative samples.
After DBSCAN clustering, the unlabeled dataset generate pseudo labels by DBSCAN algorithm. We initialize the multi-instance proxies (MIP) with randomly selected features to generate pseudo labels. We then initialize the discrepant cluster proxies (DCP) with cluster centroids of these features and use DBSCAN to perform discrepant cluster proxies collaborating and global hard negative sample mining, respectively. Then, different update designs are applied to encoder-encoded features to update DCP, and all instance features encoded by \( f_{\theta_m} \) are used to update MIP.

3. Method

3.1. Overview

Given an unlabeled person Re-ID dataset \( D = \{x_i\}_{i=1}^{N_D} \), where \( x_i \) is the i-th image and \( N_D \) is the number of images. For the USL Re-ID task, the objective is to train a robust model \( \theta \) to project a sample \( x_i \) in the data space \( D \) to a feature \( f_{\theta}(x_i) \) in the embedding space \( \mathcal{F} \).

Recently, most unsupervised Re-ID methods \([15, 9, 46, 36, 2]\) generate pseudo labels by DBSCAN algorithm. After DBSCAN clustering, the unlabeled dataset \( D \) becomes \( D' = \{(x_i, y_i)\}_{i=1}^{N_D} \), where \( y_i \in \{1, 2, \ldots, C\} \) is the pseudo label of the i-th image. \( N_D \) is the number of images after discarding outliers and \( C \) is the number of clusters. Then a memory bank \( M \) is constructed to store proxies for clusters. Since the cluster centroid contains average information, recent methods \([9, 46]\) simply use it as the uni-proxy for a cluster. Based on the proxies, the InfoNCE loss function \([32]\) is applied for model optimization. Despite there are also different variants of proxies \([15, 53, 36]\), we summarize their general formulation as follows:

\[
\mathcal{L}_{lnf_{\theta}} = -\log \frac{\exp(q \cdot p^+ / \tau)}{\sum_{i=1}^{N} \exp(q \cdot p_i / \tau)},
\]

where \( q \) is a query instance feature extracted by \( f_{\theta} \), \( p_i \) is the i-th proxy of selected \( N \) proxies from the memory bank \( M \). Among the \( N \) proxies, \( p^+ \) shares the same pseudo label with \( q \). \( \tau \) is a temperature factor. Since both \( q \) and \( p_i \) are \( L_2 \)-normalized, the cosine similarity \( q \cdot p_i \) is used as the similarity score between features.

When the model parameters are updated by gradient descent, the proxy \( p^+ \) are also updated by the query \( q \):

\[
p^+ \leftarrow \mu \cdot p^+ + (1 - \mu) \cdot q,
\]

where \( \mu \) is a momentum factor.

In this paper, we propose a contrastive learning framework based on discrepant cluster proxies and multi-instance proxies (DCMIP) as shown in Figure 3. As above, we extract the features of the training set by encoder \( f_{\theta} \) and generate pseudo labels through DBSCAN. The difference is that we simultaneously maintain cluster proxies and multi-instance proxies for a cluster, and construct contrastive loss at both the cluster and instance levels.

Due to the large number of instance proxies, we introduce a momentum encoder \( f_{\theta_m} \) following MoCo \([17]\) to maintain the consistency of negative instance proxies. The update of the momentum encoder is formulated as follows:

\[
\theta_m^t = \alpha \theta_m^{t-1} + (1 - \alpha) \theta^t,
\]

where \( \alpha \) is the momentum coefficient that controls the updated speed and is set to 0.999. The momentum encoder \( f_{\theta_m} \) evolves more smoothly, so the instance features encoded by \( f_{\theta_m} \) are more consistent. Note that, the cluster proxies are initialized and updated with the encoder-encoded features, while the instance proxies are initialized and updated with instance features encoded by \( f_{\theta_m} \).

3.2. Discrepant Cluster Proxies

We argue that the cluster uni-proxy tends to focus on the common information of a class and fails to reflect the intra-
class variance that exists. To solve this problem, we propose to maintain discrepant cluster proxies (DCP) to complementarily represent a cluster and improve the compactness of the cluster based on these discrepant proxies.

**Memory initialization.** For each cluster, we maintain $M$ cluster proxies in the memory bank $M$. For all proxies of the $j$-th cluster, we initialize them with the cluster centroid

\[ c_j = \frac{1}{|\mathcal{H}_j|} \sum_{x_i \in \mathcal{H}_j} x_i, \]

where $\mathcal{H}_j$ denotes the $j$-th cluster and $| \cdot |$ denotes the number of instances in it. Thus, the memory bank $M \in \mathbb{R}^{C \times M \times d}$ has $C \times M$ entries, and $d$ is the dimension of the features.

**Memory update.** Previous studies [22, 55] found that the dimension of the features.

\[ \partial \mathcal{L}_{Neo} \bigg/ \partial q = \frac{-1}{\tau} \left( (1 - \mathcal{P}^+) \cdot \mathcal{P}^+ - \sum_{p \in \mathcal{N}_q} \mathcal{P}^- \cdot \mathcal{P}^- \right), \]

where $\mathcal{P}^{+/-} \in [0, 1]$ is the matching probability distribution between query $q$ and the positive/negative proxy $p^+/p^-$, i.e., $\mathcal{P}^{+/-} = \frac{\exp(\mathbf{q} \cdot \mathbf{p}^{+/-}/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{q} \cdot \mathbf{p}_j/\tau)}$. $\mathcal{N}_q$ denotes the set of $N - 1$ negative proxies other than positive $p^+$. We can find that a hard positive sample with low similarity to the query tends to produce a larger gradient, generating a stronger pull to draw the query closer. But only employing such a single proxy to represent a cluster is biased and may affect the learning of inter-class relationships. Therefore, we propose to use several discrepant proxies for a cluster.

To obtain discrepant cluster proxies, we momentum update each of the $M$ initially initialized proxies of a cluster as Eq. 2 by different feature vectors from the current mini-batch. For the $m$-th proxy $p_{i,m}$ of the $i$-th cluster, the feature vector can be obtained in several ways:

\[ q_{mean} \leftarrow \frac{1}{K} \sum_{q \in Q^i} q, \]
\[ q_{rand} \leftarrow q_j, j \in Q^i, \]
\[ q_{hard} \leftarrow \arg \min_{q} q \cdot p_{i,m}, q \in Q^i, \]

where $Q^i$ is the sample feature set of the $i$-th cluster in current mini-batch. $q_{mean}$ is the average feature of the set. $q_{rand}$ is a randomly selected sample feature from $Q^i$. The selection probability is $p_{q_{rand} = q_j} = \frac{1}{K}, j = 1, 2, \ldots, K$, where $K$ denotes the number of samples for an identity in the batch. $q_{hard}$ is the sample feature which has lowest similarity with proxy $p_{i,m}$. The three different vectors correspond to three different update designs for cluster proxies, which we name “Mean”, “Rand” and “Hard”, respectively.

In our experiments, we find that the optimum cluster proxies obtained by different update designs should not only be discrepant but also stable. The discrepancy of proxies ensures the hardness of positive samples, i.e., the strength of the generated pull to the queries. The stability ensures that the pull direction of a proxy does not change drastically, otherwise, a proxy cannot form a stable pull, and a stable collaboration cannot be formed among several proxies. According to experimental results, maintaining two cluster proxies with update designs of “Mean”+“Hard” and “Mean”+“Rand” for Market-1501 and MSMT17 delivers the best performance by making a trading-off between high discrepancy and high stability. We further discuss the discrepancy and stability in Sec. 4.4.

**Cluster contrastive loss.** With $M$ discrepant cluster proxies, we form a cluster contrastive loss as follows:

\[ \mathcal{L}_{DCP} = -\frac{1}{M} \sum_{j=1}^{M} \log \frac{\exp (\mathbf{q} \cdot \mathbf{p}^j/\tau)}{\sum_{i=1}^{C} \exp (\mathbf{q} \cdot \mathbf{p}_{i,j}/\tau)}, \]

where $p_{i,j}$ is the $j$-th proxy of the $i$-th cluster. $\mathbf{p}^j$ shares the same label with the query $\mathbf{q}$ and is the $j$-th proxy for that cluster. Note that the same update design is adopted for the $j$-th proxy of all clusters.

Several discrepant proxies complementarily represent a cluster, and collaboratively reduce intra-class variance to make the cluster compact.

### 3.3. Multi-Instance Proxies

Considering that the discrepant cluster proxies cannot reflect the valuable fine-grained information contained in the hard instances of the cluster, we further maintain multi-instance proxies (MIP) for each cluster to perform global hard negative sample mining.

**Memory initialization.** We randomly select $K$ instance features encoded by the momentum encoder $f_{\theta_m}$ to initialize multi-instance proxies for each cluster. Note that $K$ equals the number of images sampled for an identity in a mini-batch. Combining cluster proxies and instance proxies, the memory bank $M \in \mathbb{R}^{C \times (M+K) \times d}$ has $C \times (M + K)$ entries in total.

**Memory update.** While updating the model parameters, the instance features of the current mini-batch are used to update the instance proxies as follows:

\[ P^i \leftarrow Q^i_m, \]

where $Q^i_m$ is the instance feature set of the $i$-th cluster in the mini-batch encoded by $f_{\theta_m}$ and $P^i$ is the set of instance proxies of that cluster in the memory bank $M$. Unlike the momentum update of the cluster proxies, the instance proxies are directly replaced by the $K$ instances with the same label in the current mini-batch. This allows us to keep as many up-to-date instance proxies as possible to represent the fine-grained information of a cluster.
Instance contrastive loss. We compute the pairwise similarity of an input query to all instance proxies of other classes in the memory bank and rank them in descending order. We select the top-$N$ most similar instance proxies as the global hardest negatives. Considering that the instance features in the current mini-batch are more up-to-date than those in the memory bank $M$, and that the momentum encoder $f_{\theta_m}$ is more stable and more robust to label noise, we choose the instance feature in the batch encoded by $f_{\theta_m}$ with the lowest similarity to the query as the hard positive. Based on the hard positive and the $N$ global hardest negatives, the following instance contrastive loss is constructed:

$$L_{MIP} = -\log \frac{\exp (q \cdot m^+) / \tau}{\sum_{i=1}^{N} \exp (q \cdot p_i^+) / \tau},$$

where $m^+$ is the hard positive and $p_i^+$ is the $i$-th hard negative instance proxies. These hard negatives accurately increase the inter-class variance of indistinguishable classes in the global embedding space from the perspective of inter-instance relationships.

### 3.4. Overall Loss

We name the contrast learning framework based on discrepant cluster proxies and multi-instance proxies DCMIP. The overall loss function of DCMIP is:

$$L_{DCMIP} = \begin{cases} L_{DCP}, & \text{if epoch} \leq E_{ins} \\ \lambda L_{DCP} + (1-\lambda) L_{MIP} & \text{else} \end{cases},$$

where $\lambda$ is the loss weight. For $L_{MIP}$, due to the poor quality of the representations in the early training stage, the hard samples at this point may be meaningless. Using these hard samples may lead to the model being trained in the wrong direction from the beginning [49]. Therefore, we set $E_{ins} = 20$ to start the instance-level contrastive learning from the 21st epoch, and the parameters of $f_{\theta_m}$ are initialized with the parameters of current $f_{\theta}$. We also report the results starting from other epochs in Appendix A.1.

DCMIP enhances the quality of representations from both intra-class and inter-class relationships. The intra-class variance is reduced by using the cluster proxies as hard positive samples in cluster contrastive loss (Eq. 8), and the inter-class variance is increased by using the instance proxies as hard negative samples in instance contrastive loss (Eq. 10). This allows the model to learn discriminative features and in turn improve the clustering quality.

### 4. Experiments

#### 4.1. Datasets and Evaluation Protocols

We evaluate our method on Market-1501 [47] and MSMT17 [38]. Market-1501 is collected with 6 cameras on the Tsinghua University campus and consists of 32,668 images of 1,501 person identities, with a training set of 12,936 images of 751 identities and a test set of 19,732 images of 750 identities. MSMT17 is a more challenging dataset, using 15 cameras for data collection and consisting of 126,441 images of 4,101 identities, with a training set of 32,621 images of 1,041 identities and a test set of 93,820 images of 3,060 identities. Both Cumulative Matching Characteristics (CMC) Top-1, Top-5, Top-10 accuracies and mean Average Precision (mAP) are adopted in our experiments.

#### 4.2. Implementation Details

We adopt ResNet50 [18] pre-trained on ImageNet [10] as the backbone. Following Cluster-Contrast [9], the generalized mean pooling [30] is used for the final pooling layer. The input image size is $320 \times 128$. At the beginning of each epoch, we use DBSCAN clustering to generate pseudo labels. The maximum distance between two samples in DBSCAN is set to 0.45 for Market-1501 and 0.7 for MSMT17. The mini-batch size is 256 consisting of 16 identities and 16 images for each identity. From the 21st epoch, we start instance-level contrastive learning and $K = 16$ instance proxies are maintained for each cluster. In instance contrastive loss (Eq. 10), we select $N = 256$ negative instance proxies for each query and set the loss weight $\lambda = 0.5$ (Eq. 11). The update momentum $\mu$ of cluster proxies is set to 0.1 (Eq. 2). The temperature hyper-parameter $\tau$ in the two losses (Eq. 8, Eq. 10) is set to 0.05. We use an Adam [23] optimizer with weight decay of $5 \times 10^{-4}$. The initial learning rate is set to $3.5 \times 10^{-5}$ and divided by 10 every 20 epochs. For both datasets, we train 50 epochs. After training, the momentum encoder $f_{\theta_m}$ is used for inference. We also provide the analysis for the maximum distance of DBSCAN and the loss weight $\lambda$ in Appendix A.1.

#### 4.3. Ablation Study

In this subsection, to analyze the effectiveness of the proposed components, we conduct extensive experiments on Market-1501 and MSMT17. We adopt the method that uses the cluster centroid as the uni-proxy of a cluster and updates the uni-proxy by the design of “Mean” as our baseline.

**Effectiveness of the discrepant cluster proxies (DCP).** Note that for Market-1501 and MSMT17, we maintain two discrepant cluster proxies and use the update designs of “Mean”+“Hard” (Eq. 5, Eq. 7) and “Mean”+“Rand” (Eq. 5, Eq. 6), respectively. As shown in Table 1, our DCP significantly exceeds the baseline using the uni-proxy, especially +4.8%/+1.5% mAP/top-1 improvement on Market-1501 and +3.2%/+2.7% mAP/top-1 improvement on MSMT17. It demonstrates that complementary and collaborative discrepant cluster proxies can describe clusters more comprehensively, therefore contributing to learning good sample representations more than the cluster uni-proxy.
### Table 1. Ablation studies on proposed components of DCMIP.

<table>
<thead>
<tr>
<th>Method</th>
<th>Market-1501</th>
<th>MSMT17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP top-1</td>
<td>mAP top-1</td>
</tr>
<tr>
<td>Baseline</td>
<td>81.0</td>
<td>35.2</td>
</tr>
<tr>
<td>Baseline+DCP</td>
<td>85.8</td>
<td>38.4</td>
</tr>
<tr>
<td>Baseline+MIP</td>
<td>84.5</td>
<td>39.2</td>
</tr>
<tr>
<td>DCMIP</td>
<td><strong>86.7</strong></td>
<td><strong>40.9</strong></td>
</tr>
</tbody>
</table>

### Table 2. Ablation studies on different hard sample mining techniques. In all rows, we keep the weight $\lambda$ of cluster contrastive loss in total loss (Eq. 11) as 0.5 from the 21st epoch.

<table>
<thead>
<tr>
<th>Method</th>
<th>Market-1501</th>
<th>MSMT17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP top-1</td>
<td>mAP top-1</td>
</tr>
<tr>
<td>DCMIP w/o MIP</td>
<td>85.4</td>
<td>37.5</td>
</tr>
<tr>
<td>+ MIP</td>
<td><strong>86.7</strong></td>
<td><strong>40.9</strong></td>
</tr>
<tr>
<td>+ Batch hard triplet</td>
<td>86.1</td>
<td>37.9</td>
</tr>
<tr>
<td>+ Batch hard instance</td>
<td>86.0</td>
<td>38.4</td>
</tr>
</tbody>
</table>

### Effectiveness of the multi-instance proxies (MIP).

To demonstrate the effectiveness of MIP, we combine MIP with the baseline and DCP respectively. In Table 1, comparing to the baseline, mAP/top-1 of Baseline+MIP is improved by 3.5%/1.1% on Market-1501 and 4.0%/2.2% on MSMT17. DCMIP (DCP+MIP) increases mAP/top-1 of Baseline+DCP by 0.9%/0.4% and 2.5%/0.5% on Market-1501 and MSMT17 severally. This demonstrates that for both cluster uni-proxy and multi-proxies, global hard negative mining based on MIP can capture the fine-grained information contained in truly hard instances in the global embedding space. In Table 2, we compare MIP with two batch hard sample mining techniques. One way is hard batch triplet mining [20], which forms a triplet with the anchor, the hardest positive, and the hardest negative in the mini-batch. The other way is batch hard instance mining [2], which uses the most similar instance of the same class and all instances of other classes in the mini-batch as the positive and the negatives. As the results show, global hard negative sample mining based on MIP outperforms the two techniques. This demonstrates that our MIP overcomes the limitation of batch hard sample mining by exploiting the hardest negative instance proxies to purposefully increase the inter-class variance of indistinguishable classes.

DCMIP combines DCP and MIP for contrastive learning based on both cluster proxies and instance proxies. Compared with the cluster uni-proxy baseline, our method improves mAP/top-1 by 5.7%/1.9% on Market-1501 and 5.7%/3.2% on MSMT17 by a large margin. We believe that DCMIP can reduce intra-class variance through the collaboration of discrepant cluster proxies and increase inter-class variance through global hard negative mining based on multi-instance proxies.

### Table 3. Comparison of different update policies for cluster proxies.

<table>
<thead>
<tr>
<th>Update policy</th>
<th>Market-1501</th>
<th>MSMT17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP top-1</td>
<td>mAP top-1</td>
</tr>
<tr>
<td>Mean</td>
<td>81.0</td>
<td>35.2</td>
</tr>
<tr>
<td>Rand</td>
<td>81.6</td>
<td>35.8</td>
</tr>
<tr>
<td>Hard</td>
<td>82.6</td>
<td>38.4</td>
</tr>
<tr>
<td>Mean+Hard</td>
<td><strong>85.8</strong></td>
<td><strong>38.4</strong></td>
</tr>
<tr>
<td>Mean+Rand</td>
<td>83.1</td>
<td>31.8</td>
</tr>
<tr>
<td>Rand+Hard</td>
<td>84.7</td>
<td>34.8</td>
</tr>
<tr>
<td>Mean+Rand+Hard</td>
<td>85.5</td>
<td>37.5</td>
</tr>
</tbody>
</table>

### Clustering quality.

To intuitively demonstrate the ability of our method to reduce intra-class variation and inter-class similarity, we visualized randomly selected samples of 20 classes by t-SNE [33]. As shown in Figure 4, the compactness of all classes is significantly improved in DCMIP compared to the baseline. For several classes that are too close to distinguish in the baseline, our method increases their inter-class distances. Moreover, for the two classes with mixed features, DCMIP successfully separates them. We also report the results of cluster quality measured with four cluster evaluation metrics on Market-1501 and MSMT17 in Appendix A.2.

### 4.4. Parameter Analysis

#### Different update policies for cluster proxies.

We defined three update designs in Sec. 3.2 to update the cluster proxies: “Mean”, “Rand”, and “Hard”, as shown in Eq. 5, Eq. 6, and Eq. 7, respectively. Several cluster proxies are obtained by applying different update designs to the same initial cluster centroid. The three designs can form seven different update policies by combination: “Mean”, “Rand”, “Hard”, “Mean+Hard”, “Mean+Rand”, “Rand+Hard”, and “Mean+Rand+Hard”. As shown in Table 3, discrepant cluster proxies obtained by appropriate update policies outperform the uni-proxy, but the number of cluster proxies

---

**Figure 4.** The t-SNE visualization of 20 random classes in Market-1501 between baseline and our DCMIP. Different colors represent different IDs. At the bottom, we show the case of small inter-class variance and large intra-class variance through partial person images of several identities.
Complements the stability of “Hard”, thus achieving the best balance between high discrepancy and high stability. For MSMT17, although “Mean+Rand” has a low discrepancy, it avoids the problem of “Hard” and forms discrepant proxies which can collaborate stably to achieve the best performance. We conjecture that since three dynamically changing proxies are more difficult to form a stable collaboration than two proxies, despite the high discrepancy and the ability to represent more information, the policy “Mean+Rand+Hard” is not optimal.

The number of hard negative instance proxies. We analyze the number of hard negative instance proxies $N$ selected by global hard negative mining. In Figure 6, we can see that the performance of Market-1501 and MSMT17 firstly increases and then decreases as $N$ raise. $N=0$ indicates the case of cluster-level contrastive learning based on DCP only. Both datasets achieve the best mAP when we set $N$ to 256. We speculate that when $N>0$, we can effectively increase distances between classes hard to distinguish with the valuable globally hardest negative samples. However, as $N$ grows, meaningless easy samples may be selected, instead reducing the matching probability of meaningful samples and affecting the gradient thus causing a decrease in performance. Therefore, we set $N'=256$.

4.5. Comparison with State-of-the-Arts

In Table 4, we compare our DCMIP with state-of-the-art Re-ID methods on Market-1501 and MSMT17. In an unsupervised setting, our DCMIP significantly outperforms previous methods. We achieve mAP/top-1 of 86.7%/94.7% and 40.9%/69.3% on Market-1501 and MSMT17, respectively. Compared to unsupervised methods without any labels, our discrepant cluster proxies and multi-instance proxies substantially improve mAP by 3.7% and 7.9% on Market-1501 and MSMT17 than the uni-proxy method Cluster-Contrast [9]. Moreover, our DCMIP surpasses the second-best method ISE [46] on Market-1501 and MSMT17 by 1.4% and 3.9% in mAP and outperforms ICE [2] and PPLR [7] on MSMT17 by a remarkable margin.

Compared to unsupervised methods with camera labels, our DCMIP without any camera knowledge outperforms four...
Table 4. Comparison with state-of-the-art methods on Market-1501 and MSMT17. The best results of unsupervised methods without any labels are marked in **bold**. Note that the input image size of DCMIP is 320 × 128.

<table>
<thead>
<tr>
<th>Method</th>
<th>Market-1501</th>
<th></th>
<th>MSMT17</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>top-1</td>
<td>top-5</td>
<td>top-10</td>
</tr>
<tr>
<td><strong>Unsupervised methods with camera labels</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IICS [41]</td>
<td>CVPR’21</td>
<td>72.9</td>
<td>89.5</td>
<td>95.2</td>
</tr>
<tr>
<td>CAP [36]</td>
<td>AAAI’21</td>
<td>79.2</td>
<td>91.4</td>
<td>96.3</td>
</tr>
<tr>
<td>ICE [2]</td>
<td>ICCV’21</td>
<td>82.3</td>
<td>93.8</td>
<td>97.6</td>
</tr>
<tr>
<td>PPLR [7]</td>
<td>CVPR’22</td>
<td>84.4</td>
<td>94.3</td>
<td>97.8</td>
</tr>
<tr>
<td><strong>Unsupervised methods without any labels</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUC [27]</td>
<td>AAAI’19</td>
<td>29.6</td>
<td>61.9</td>
<td>73.5</td>
</tr>
<tr>
<td>JVTC [24]</td>
<td>ECCV’20</td>
<td>41.8</td>
<td>72.9</td>
<td>84.2</td>
</tr>
<tr>
<td>MMCL [34]</td>
<td>CVPR’20</td>
<td>45.5</td>
<td>80.3</td>
<td>89.4</td>
</tr>
<tr>
<td>HCT [43]</td>
<td>CVPR’20</td>
<td>56.4</td>
<td>80.0</td>
<td>91.6</td>
</tr>
<tr>
<td>SpC [15]</td>
<td>NeurIPS’20</td>
<td>73.1</td>
<td>88.1</td>
<td>95.1</td>
</tr>
<tr>
<td>JVTC++ [3]</td>
<td>CVPR’21</td>
<td>75.4</td>
<td>90.5</td>
<td>96.2</td>
</tr>
<tr>
<td>OPLG-HCD [50]</td>
<td>ICCV’21</td>
<td>78.1</td>
<td>91.1</td>
<td>96.4</td>
</tr>
<tr>
<td>ICE [2]</td>
<td>ICCV’21</td>
<td>79.5</td>
<td>92.0</td>
<td>97.0</td>
</tr>
<tr>
<td>MCRN [39]</td>
<td>AAAI’22</td>
<td>80.8</td>
<td>92.5</td>
<td>-</td>
</tr>
<tr>
<td>SECRET [19]</td>
<td>AAAI’22</td>
<td>81.0</td>
<td>92.6</td>
<td>-</td>
</tr>
<tr>
<td>PPLR [7]</td>
<td>CVPR’22</td>
<td>81.5</td>
<td>92.8</td>
<td>97.1</td>
</tr>
<tr>
<td>RMCL [29]</td>
<td>KBS’23</td>
<td>81.7</td>
<td>93.0</td>
<td>97.6</td>
</tr>
<tr>
<td>Cluster-Contrast [9]</td>
<td>ACCV’22</td>
<td>83.0</td>
<td>92.9</td>
<td>97.2</td>
</tr>
<tr>
<td>ISE [46]</td>
<td>CVPR’22</td>
<td>85.3</td>
<td>94.3</td>
<td><strong>98.0</strong></td>
</tr>
<tr>
<td><strong>DCMIP (320 × 128)</strong></td>
<td>This paper</td>
<td><strong>86.7</strong></td>
<td><strong>94.7</strong></td>
<td><strong>98.0</strong></td>
</tr>
</tbody>
</table>

Methods (i.e., IICS [41], CAP [36], ICE [2], PPLR [7]) on Market-1501 and three (i.e., IICS [41], CAP [36], ICE [2]) on MSMT17 in mAP. In addition, under the supervised setting, our DCMIP achieves competitive performance to the well-known supervised method DG-Net [51] and ADB-Net [4]. It is worth noting that DCMIP with ground truth scores higher in mAP and top-1 than ISE [46] by 11.8% and 7.1% on MSMT17, which demonstrates the superiority and potential of our approach on large datasets.

5. Discussion

Our DCMIP reduces intra-class variation by two discrepant cluster proxies for all clusters, but this may not be the optimal solution. For clusters with high intra-class compactness, further reducing the intra-class variation is not necessary, as it may impair generalizability. For clusters with low intra-class compactness, more cluster proxies are required to represent diverse subsets and lower intra-class variance. We will explore additional strategies to obtain discrepant proxies as well as a dynamic cluster proxy number for different clusters in the future study.

6. Conclusion

In this paper, we propose a contrastive learning framework based on discrepant cluster proxies and multi-instance proxies for unsupervised person re-identification. We maintain two discrepant cluster proxies by different update designs to complementarily represent a cluster and act as hard positive samples in the cluster contrastive loss to collaboratively reduce intra-class variance. We also maintain multi-instance proxies for a cluster to accurately represent the fine-grained instance information. Then global hard negative sample mining is performed among the instance proxies to increase the inter-class variance of indistinguishable classes through the instance contrastive loss. Comprehensive experiments have shown that our framework outperforms prior state-of-art methods on two prevalent datasets.

Acknowledgment

This work was supported by the National Key Research and Development Project of New Generation Artificial Intelligence of China under Grant 2018AAA0102504. We also thank Xueming Qian for valuable advice.
References


