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

SUPPLEMENTARY MATERIAL

Chang Zou[1] Zeqi Chen[2] Zhichao Cui[3] Yuehu Liu[2*] Chi Zhang[2]
[1]School of Software Engineering, Xi'an Jiaotong University
[2]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University [3]Chang'an University

In this supplementary material, we demonstrate more experimental results and implementation details of our method.

A. More Experimental Results

A.1. Parameter Analysis

The cluster contrastive loss weight λ. We analyze the weight λ of cluster contrastive loss $\mathcal{L}_{DCP}$ (Eq. 8) in the overall loss $\mathcal{L}_{DCMIP}$ (Eq. 10) on Market-1501 and MSMT17 in Figure 1. λ controls the proportion of the cluster contrastive loss and instance contrastive loss. When λ is small and the instance contrastive loss is weighted more heavily, the performance on both datasets drops significantly, especially for MSMT17. However, when λ is large and the cluster contrastive loss is weighted more heavily, the model still achieves good performance. This suggests that, even after the inclusion of instance-level contrastive learning, cluster-level contrastive learning still contributes more to performance, and that the inter-instance relationships learned based on multi-instance proxies are complementary to the inter-class relationships learned based on discrepant cluster proxies. We set $\lambda = 0.5$ because the model achieves the best performance on both datasets at that value.

The distance threshold for DBSCAN clustering. In DBSCAN [3], the clustering threshold is the maximum distance that two samples can have from one another and still be considered neighbors. A larger distance threshold may result in samples with the same ground truth being incorrectly merged, while a smaller distance threshold may result in incorrect splits. Figure 2 shows the sensitivity of our DCMIP to the distance threshold. We find that the smaller threshold is more suitable for the relatively small dataset Market-1501, while the larger threshold is more suitable for the relatively large MSMT17. The optimum distance threshold on Market-1501 and MSMT17 is 0.45 and 0.7, respectively. Although different methods may have different optimal thresholds on the same dataset, this conclusion is consistent with recent works. The state-of-arts method PPLR [1] set the threshold to 0.6 for Market-1501 and 0.7 for MSMT17, and ISE [4] set it to 0.4 for Market-1501 and 0.7 for MSMT17.

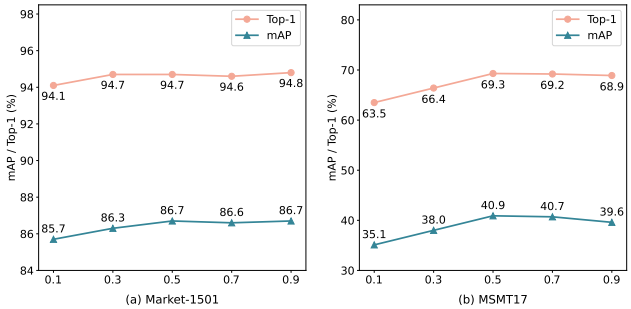

Figure 1. Parameter analysis of the loss weight λ on Market-1501 and MSMT17.

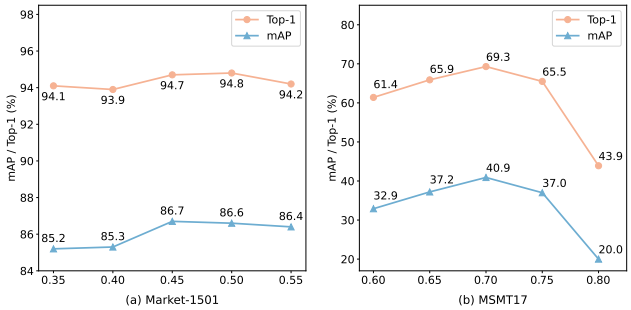

Figure 2. Parameter analysis of the distance threshold in DBSCAN clustering on Market-1501 and MSMT17.

The start epoch for instance-level contrastive learning. Considering the poor quality of the representations learned by the model in the early training stage and that the hard samples at this point may be meaningless [5], we specify to maintain multi-instance proxies and perform global hard sample mining from the 21st epoch (*i.e.*, $E_{ins} = 20$). We also analyze the case of starting from the 1st epoch, the 11th epoch, the 31st epoch, and the 41st epoch on Market-1501 and MSMT17. As shown in Table 1, compared with the case of only performing cluster-level contrastive learning

*Corresponding author

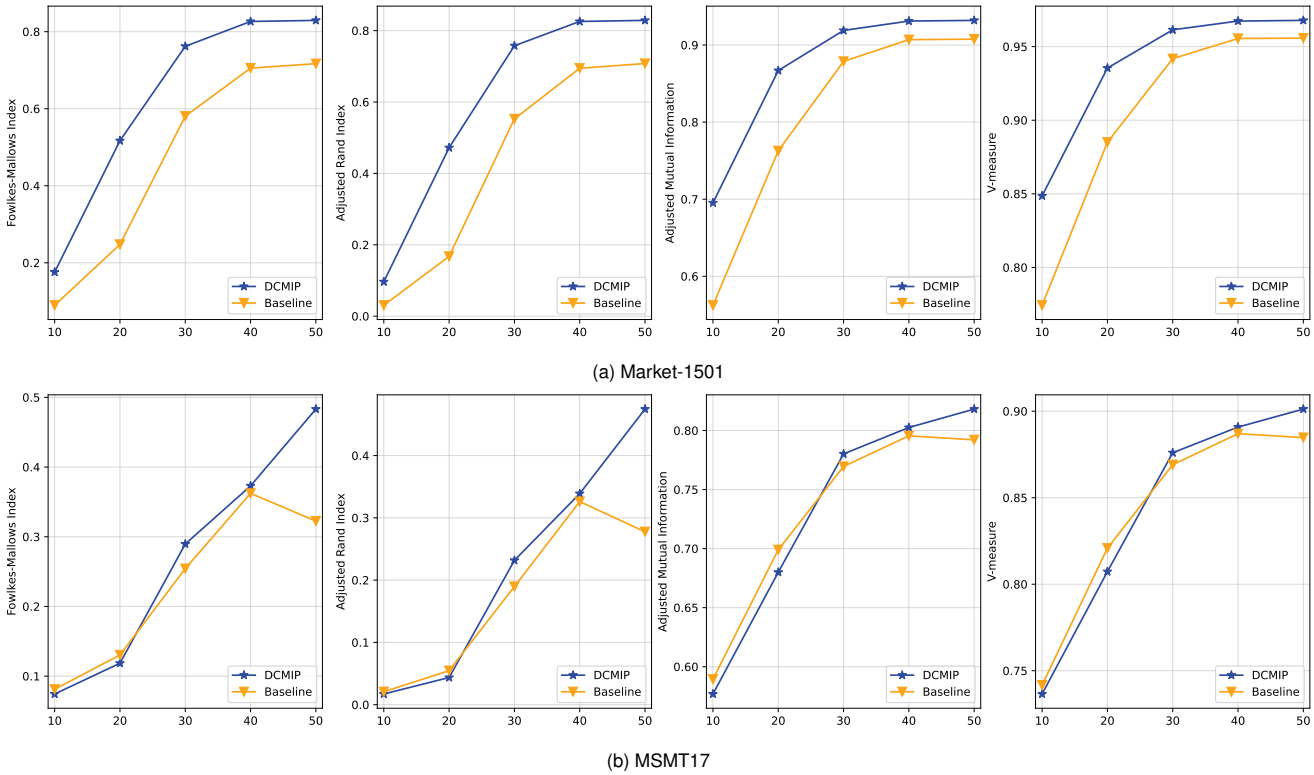

Figure 3. Four clustering evaluation metrics over different epochs for the baseline and our DCMIP on (a) Market-1501 and (b) MSMT17 datasets.

Start epoch	Market-1501		MSMT17	
	mAP	top-1	mAP	top-1
DCMIP w/o MIP	85.4	93.7	37.5	68.0
Epoch 1	80.1	91.3	36.1	64.2
Epoch 11	84.3	92.8	37.7	65.7
Epoch 21	**86.7**	**94.7**	**40.9**	**69.3**
Epoch 31	86.4	**94.7**	39.7	68.2
Epoch 41	86.1	94.3	39.3	68.1

Table 1. Parameter analysis on the start epoch of maintaining multi-instance proxies and global hard negative sample mining.

throughout the whole training process (DCMIP w/o MIP), the mAP and top-1 decrease on both datasets when start ing from the 1st epoch and the 11th epoch. Moreover, the accuracy of beginning from epoch 1 decreases more significantly than from epoch 11. This indicates that the features learned in the early training stage are indeed unreliable and the hard samples are meaningless. The emphasis on these samples results in the wrong optimization direction at the beginning, which leads to performance degradation. Global hard sample mining based on multi-instance proxies starting from epoch 21, epoch 31, and epoch 41 improve performance to varying degrees. We speculate that this is because the embedding space structure learned by the model in the middle and late training stages already has a certain degree of reliability and the hard samples mined at this time are truly informative samples. In addition, we compare choosing the hardest, semi-hard, easiest positive and negative samples from epoch 21 in Table 2. The results shows that the benefits of the hardest samples outweigh the impact of label noise in the middle and later training stage. Choosing less hard samples to avoid noise will instead miss some valuable information.

Hardness of samples	Market-1501		MSMT17	
	mAP	top-1	mAP	top-1
Hardest	**86.7**	**94.7**	**40.9**	**69.3**
Semi-hard	86.3	94.4	39.1	69.0
Easiest	85.8	94.3	38.0	67.9

Table 2. Comparison of the hardness of positive and negative samples in the instance contrastive loss.

Method	Market-1501		MSMT17	
	mAP	top-1	mAP	top-1
DCMIP w/ f_{θ_m}	**86.7**	**94.7**	**40.9**	**69.3**
DCMIP w/o f_{θ_m}	81.7	92.1	33.4	60.3

Table 3. Comparison of with and without the momentum encoder f_{θ_m}.

The effectiveness of the momentum encoder. We maintain $K = 16$ instance proxies for each cluster. Considering the large number of instance proxies and the fact that only a

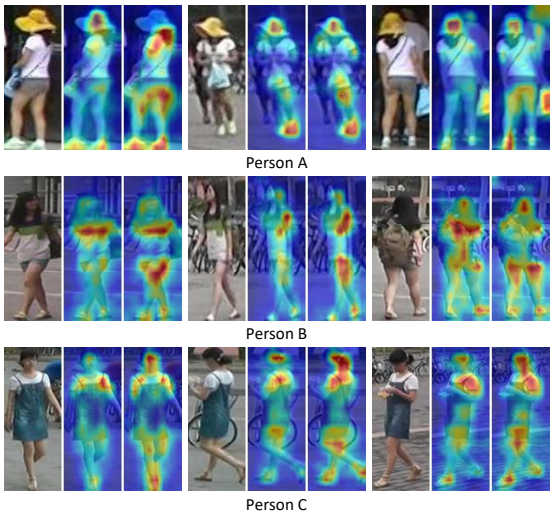

Figure 4. Each triplet of a person includes, from left to right, the original image, action map of DCMIP, and action map of the cluster uni-proxy baseline. These images suggest that our DCMIP prefers to focus on discriminative visual cues (*e.g.*, hat, bag, and clothing patterns) and treats other cues (*e.g.*, legs) as intra-class commonalities.

small fraction of classes can be updated in an iteration, we follow MoCo to maintain the consistency of the negative instance proxies with the momentum encoder f_{θ_m}. In Table 3, we compare the cases with and without the momentum encoder on Market-1501 and MSMT17. When without the momentum encoder, the instance proxies are initialized and maintained by the encoder. The experimental results show that not using the momentum encoder leads to significant performance degradation on both datasets, so maintaining the consistency of negative instance proxies is essential.

A.2. Clustering Quality

As shown in Figure 3, we evaluate the clustering quality of model features on Market-1501 and MSMT17 during the the training process for baseline and our DCMIP, respectively. We use four clustering evaluation metrics from [2]: Fowlkes-Mallows index, Adjusted Rand index, Adjusted Mutual Information, and Adjusted Mutual Information. For all metrics, the higher the value, the better the clustering quality. The experimental results show that our method is effective in improving the cluster quality on both datasets and helps the model learn more discriminative representations. In addition, the clustering quality of Market-1501 is significantly higher than that of MSMT17.

A.3. Qualitative analysis

Figure 4 demonstrates the activation maps of baseline and DCMIP on Market-1501. Compared to the baseline, for the same person, the model of DCMIP tends to focus on cues that distinguish the person from the others (*e.g.*, hat, bag, and clothing patterns), while cues that are not highly discriminative (*e.g.*, legs) are considered as a commonality of the person and are not emphasized. Therefore, our DCMIP is more beneficial to help the model learn representations with high intra-class similarity and high inter-class variance.

B. Algorithm Details

The algorithm details are provided in Algorithm 1.

Algorithm 1: Discrepant and multi-instance proxies for purely unsupervised person Re-ID.

Input : Unlabeled training set $\mathcal{D}$, an encoder f_θ, a momentum encoder f_{θ_m}, max epoch E_{max}, max iterations I_{max}, the start epoch E_{ins} to maintain MIP, loss weight λ, the update policy.
Output: Trained momentum encoder f_{θ_m}.

for $epoch = 1$ **to** E_{max} **do**
  DBSCAN cluster on features encoded by f_θ and get pseudo labeled dataset $\mathcal{D}'$;
  Construct a memory bank $\mathcal{M}$ with cluster centroids as cluster proxies;
  if $epoch > E_{ins}$ **then**
    Initialize f_{θ_m} with the parameters of f_θ;
    Add instance features encoded by f_{θ_m} to $\mathcal{M}$ as instance proxies for each cluster;
    for $iter = 1$ **to** I_{max} **do**
      Select the global top-$\mathcal{N}$ hardest negative samples from $\mathcal{M}$;
      Train f_θ with the total loss in Eq. 11;
      Update f_{θ_m} by Eq. 3 and instance proxies;
    end
  else
    for $iter = 1$ **to** I_{max} **do**
      Train f_θ with cluster contrastive loss Eq. 8;
    end
  end
  Update cluster proxies according to the update policy;
end

C. More Implementation Details

Our code is implemented based on Pytorch. Training is done on 4 NVIDIA GeForce RTX 3090 GPUs, and only one GPU is used for inference. We use random flipping, random cropping, and random erasing for data augmentation. For DBSCAN, we compute the Jaccard distance based on the k-reciprocal encoding, where k is set to 30. The minimum number of neighbors for a core point in DBSCAN is set to 4. In each epoch, we train the model for 200 iterations. Each iteration contains forward propagation, network parameter updates during backward propagation, and updates

for cluster proxies or both cluster proxies and instance proxies according to the start epoch of instance-level contrastive learning. Considering that the momentum encoder is more robust than the encoder and has better performance in experiments, we use the momentum encoder for inference.

References

[1] Yoonki Cho, Woo Jae Kim, Seunghoon Hong, and Sung-Eui Yoon. Part-based Pseudo Label Refinement for Unsupervised Person Re-identification, Mar. 2022. 1

[2] David Cournapeau and Google members. scikit-learn. https://scikit-learn.org/stable/index.html, 2007. 3

[3] 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

[4] Xinyu Zhang, Dongdong Li, Zhigang Wang, Jian Wang, Errui Ding, Javen Qinfeng Shi, Zhaoxiang Zhang, and Jingdong Wang. Implicit Sample Extension for Unsupervised Person Re-Identification, Apr. 2022. 1

[5] Wenzhao Zheng, Jiwen Lu, and Jie Zhou. Hardness-Aware Deep Metric Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(09):3214–3228, Sept. 2021. 1