Supplementary Material: Lifelong Unsupervised Domain Adaptive Person Re-identification with Coordinated Anti-forgetting and Adaptation

Zhipeng Huang†‡*, Zhizheng Zhang‡†† Cuiling Lan‡† Wenjun Zeng3 Peng Chu2 Quanzeng You2 Jiang Wang2 Zicheng Liu2 Zheng-jun Zha1
1University of Science and Technology of China
2Microsoft 3EIT Institute for Advanced Study
hzp1104@mail.ustc.edu.cn zengw2011@hotmail.com zhazj@ustc.edu.cn {zhizheng, culan, pengchu, quyou, jiangwang, zliu}@microsoft.com

In this supplementary material, we first detail our modified ID-wise reservoir sampling algorithm which is originally an instance-wise one. Then we provide detailed introduction for datasets used in this work and give more implementation details. Finally, we present more experimental results and discuss the ethical impact of this work.

1. ID-wise Reservoir Sampling Algorithm

As introduced in our main body, we set up a memory buffer with limited size for data replay, which is updated using ID-wise Reservoir Sampling algorithm at the end of each stage. Here, we detail this algorithm which is modified from the regular one that is instance-wise.

The regular reservoir sampling algorithm [8] is designed to choose a subset of \( k \) individuals at random, without replacement, from a population of size \( N \) in a sequence. Here, \( N \) is allowed to be unknown and typically large. In this algorithm, each sample is chosen with an equal probability over all identities. However, the regular reservoir sampling algorithm may suffer from over-fitting for the data replaying adopted in our targeted LUDA person ReID problem, we expect the stored samples are of diverse data statistics. Thus, we propose to modify the regular reservoir sampling algorithm to be an ID-wise one, in the sense that each ID will be randomly chosen for storing with an equal probability over all identities that have been seen so far.

We describe the ID-wise reservoir sampling algorithm in Alg. 1. The record function summarizes all samples into a dictionary structure according to their identities, where the keys of the dictionary are the recorded identities while its values are the corresponding sample indexes. The select function randomly selects \( K \) samples for a given ID. Here, if the number of samples of the given ID is less than \( K \), we will adopt data augmentation (i.e., random horizontal flipping, random cropping and random erasing) to augment the current samples and store augmented samples. The random function randomly produces an integer between the provided minimum and maximum values inclusively.

We compare it to the regular instance-wise reservoir sampling algorithm in the Fig.5 of our main body. The experimental results demonstrate the superiority of our modified ID-wise reservoir sampling algorithm relative to its original version when adopted to our proposed scheme CLUDA-ReID for LUDA person ReID.

---

*This work was done when Zhipeng Huang was an intern at MSRA.
†Equal contribution.
‡Corresponding authors.

---

**Algorithm 1** ID-wise Reservoir Sampling Algorithm

\( \mathcal{M} \): the memory buffer with the size of \( ||\mathcal{M}|| \);
\( N_{id} \): the total number of IDs observed so far;
\( K \): the number of samples stored for each identity;
\( \mathcal{X}'_j \): the data set in the \( j \)-th stage of the target stream;
\( \mathcal{Y}'_j \): the pseudo labels of \( \mathcal{X}'_j \), obtained via clustering.

\[
\text{procedure ID-RESERVOIR}(\mathcal{M}, ||\mathcal{M}||, N_{id}, K, \mathcal{X}'_j, \mathcal{Y}'_j) \\
\text{id dicts} = \text{record}(\mathcal{X}'_j, \mathcal{Y}'_j) \\
\text{for } i = 0 \text{ to len(id dicts.keys())} - 1 \text{ do} \\
\text{indices} = \text{id dicts}[i] \\
(x_k^i, y_k^i)_K = \text{select}(\mathcal{X}'_j, \mathcal{Y}'_j, \text{indices}, K) \\
\text{if } ||\mathcal{M}|| / K > N_{id} \text{ then} \\
\mathcal{M}[N_{id} * K, (N_{id} + 1) * K] \leftarrow (x_k^i, y_k^i)_K \\
\text{else} \\
r = \text{random}(\text{min} = 0, \text{max} = N_{id}) \\
\text{if } r < ||\mathcal{M}|| / K \text{ then} \\
\mathcal{M}[r * K, (r + 1) * K] \leftarrow (x_k^i, y_k^i)_K \\
\text{end if} \\
N_{id} \leftarrow N_{id} + 1 \\
\text{end for} \\
\text{return } \mathcal{M} \\
\text{end procedure}
\]
2. Optimization of Coordinated Data Replay

We describe the training procedure of CLUDA-ReID using the proposed coordinated data replay in Alg. 2. For simplicity, we show the case where we take the “task” of adaptation as meta-train while taking the “task” of anti-forgetting as meta-test. Theoretically, as indicated in the Eq.(6) of our manuscript, the orders of these two tasks are exchangeable for the meta optimization. Our empirical study also demonstrates that iteratively choosing one of these two tasks (i.e., adaptation and anti-forgetting) as meta-train while taking the other one as meta-test delivers very close performance with our reported ones.

3. Detailed Introduction for Datasets

In this work, we use four public datasets PersonX (PX) [7], Market1501 (MA) [11], CUHK-SYSU (SY) [10], MSMT17 (MS) [9] for pre-training and lifelong unsupervised domain adaptation, and build a new dataset MMP-Retrieval for unseen domain generalization evaluation.

PersonX [7] is a synthetic dataset generated based on Unity [6], containing 45,792 images, where 410 identities are used for training and 856 identities are used for testing.


CUHK-SYSU [10] is originally a large-scale dataset for person search task, containing 18,184 images and 8,432 identities. Following the previous work [5], we employ a subset of CUHK-SYSU using the bounding box annotations, in which each identity includes at least 4 person crops. As a result, the training set of this subset includes 942 identities while its test set includes 2,900 identities.

MSMT17 [9] is a large-scale person ReID dataset which is captured from 15 cameras (including 12 outdoor cameras and 3 indoor cameras). It contains 126,441 images of 4,101 identities, where 1,041 identities and 3,060 identities are used for training and testing respectively.

MMP-Retrieval, the new dataset we propose in this paper, is built upon the Multi-camera Multiple People Tracking dataset (i.e., MMPTRACK) which is released in ICCV 2021 Multi-camera Multiple People Tracking Workshop*. It is available at https://iccv2021-mmp.github.io/subpage/dataset.html. In MMP-Retrieval, all videos are collected in 5 simulated environments: retail, lobby, industry, cafe and office, including 28 people participating in recording (14 in training, 7 in validation and 7 in testing). All people were paid and signed an agreement to release their data to the public for research usage. So there is no privacy issue. MMP-Retrieval is built using the combination of the training and validation splits of MMPTRACK dataset, comprising 21 identities in all. To make the samples in MMP-Retrieval diverse, we uniformly downsample the original video sequences of MMPTRACK with a ratio of 128, then divide each downsampled sequence into two halves. We use the cropped persons in the first half as the query set and those in the second half as the gallery set. Although this dataset contains a limited number of identities, the diverse camera angles and cluttered backgrounds (as illustrated in Fig. 1) still make MMP-Retrieval a challenging test dataset for unseen-domain generalization evaluation. To avoid cloth-changing cases crossing different environments, we report the Rank-1 and mAP scores averaged over all environments in our experiments.

*https://iccv2021-mmp.github.io

Algorithm 2 Coordinated Data Replay (CDR) Algorithm

\( \mathcal{M} \): the memory buffer for storing old samples;
\( \theta \): the parameters of the backbone network;
\( \phi \): the parameters of the classifier;
\( \alpha \): the learning rate of the meta-train optimization;
\( \eta \): the learning rate of the meta-optimization;
\( L_{\text{Adap}} \): the optimization objective for adaptation;
\( L_{\text{AntiF}} \): the optimization objective for anti-forgetting.

Init: parameters \( \Psi = \{\theta, \phi\} \), learning rates \( \eta, \alpha \).
for \( i \) in iterations do
  Meta-train:
  Sample a batch \( B^n \) from the new data.
  Compute \( L_{\text{Adap}}(\Psi_t) \) with \( B^n \). \( \triangleright \) Eq. (1)
  Update \( \Psi \) w.r.t. \( L_{\text{Adap}} \):
  \( \Psi'_t \leftarrow \Psi_t - \alpha \nabla_{\Psi_t} L_{\text{Adap}}(\Psi_t) \)
  Meta-test:
  Sample a batch \( B^o \) from the old data stored in \( \mathcal{M} \).
  Compute \( L_{\text{AntiF}}(\Psi'_t) \) with \( B^o \) and \( B^o \). \( \triangleright \) Eq. (2)
  Meta optimization:
  \( L_{\text{CDR}} = L_{\text{Adap}}(\Psi_t) + L_{\text{AntiF}}(\Psi'_t) \) \( \triangleright \) Eq. (4)
  Update model parameters:
  \( \Psi_{t+1} \leftarrow \Psi_t - \eta \nabla_{\Psi_t} L_{\text{CDR}} \).
end for
Output: \( \theta, \phi \)
4. More Implementation Details

Following the prior works for UDA person ReID [2, 3], we use the clustering algorithm of DBSCAN [1] to generate pseudo labels, where the maximum distance between neighbors is set to 0.6 and the minimal number of neighbors for a dense point is set to 4. We balance different terms in $\mathcal{L}_{Adap}$ and $\mathcal{L}_{AntiF}$ to make their corresponding gradients lie in similar ranges, and we empirically find that setting all of their weights to be 1 works well. We train the model on the source domain for 60 epochs, and train the model on each domain of the stationary/dynamic scenarios for 40/60 epochs, respectively. We adopt Adam [4] optimizer with a weight decay of $5 \times 10^{-4}$. In the pre-training stage on the source domain or each fine-tuning stage on the target domain, as in Alg. 2, the learning rate $\eta$ for the meta optimization is initialized to $3.5 \times 10^{-4}$ and the learning rate $\alpha$ is initialized to $3.5 \times 10^{-3}$. Both $\eta$ and $\alpha$ is decayed with a factor of 0.1 in the 40th epoch. Unless otherwise specified, the size of the memory buffer is set to 512, and 8 randomly sampled images are stored for each sampled identity using the Alg. 1.

5. More Experiment Results

5.1. Empirical Study for the Memory Buffer Size

In this section, we conduct an empirical study for the size of the memory buffer. As shown in Fig. 2, the performances on the source and target domains both become higher as the size of the memory buffer increases. This is because more historical data will deliver more benefits for capturing new knowledge while recalling old knowledge, thanks to the coordination between timely adaptation and anti-forgetting in our proposed scheme. In practice, we need to consider the trade-off between the storage burdens and performance. Thus, we recommend using the memory size of 512.

5.2. More Results with the mAP Metric

For simplicity, in our manuscript, we only report the experiment results of our ablation study using the Rank-1 metric. In this section, we provide more experiment results using mAP as the evaluation metric in Fig.3 and Fig.4. Comparing the Fig.3 and Fig.4 in this supplementary to the Fig.4 and Fig.5 in our manuscript, we can find the experiment results in mAP has very consistent trend with those in Rank-1. This further demonstrates the effectiveness and superiority of different components in our proposed CLUDA-ReID.
In practice, the order of target domains is agnostic. To verify this, we conduct an experiment in the dynamic target scenario with another domain order: PX → MS → MA → SY. This shows the practicality of CLUDA-ReID in timely adapting to new environments.

5.3. Experimental Analysis for the Training Orders

In the manuscript, we demonstrate the effectiveness of CLUDA-ReID in the dynamic target scenario with the training order of PX → MA → SY → MS → MA → SY. In practice, the order of target domains is agnostic. To verify this, we conduct an experiment in the dynamic target scenario with another domain order: PX → MS → MA → SY → MS → MA → SY. Other configurations are kept the same as that in the Sec.5.4 of the manuscript.

In Tab.1, we measure the performance at the end of each stage for evaluating the timely adaptation performance. The scheme Base.+ CDR achieves 3.87%/1.90%, 3.63%/5.43%, 5.29%/9.92% improvements in R-1/mAP on MS, MA and SY respectively when these domains first appear, and achieves 5.91%/2.73%, 3.51%/8.88%, 10.88%/14.99% improvements in R-1/mAP on MS, MA and SY respectively when these domains show up for the second time. The scheme Base.+ CDR+ RCL is superior to the baseline Stage-wise UDA by 7.50%/3.02%, 7.75%/10.56%, 18.07%/20.14% improvements in R-1/mAP on MS, MA and SY when these domains first appear, and by 12.05%/6.08%, 12.40%/17.90%, 27.90%/29.30% improvements in R-1/mAP on MS, MA and SY when they appear for the second time. The trends of experiment results in this domain order is consistent with those in the Sec.5.4 of our manuscript, further demonstrating the effectiveness of CLUDA-ReID in timely adapting to new environments.

In Tab.2, we measure the performance at the end of the 3rd stage (i.e., t=3, all domains are traversed once) and at the end of the 6th stage (i.e., t=6, all domains are seen for the second time), for evaluating the anti-forgetting capacity of our proposed method. Relative to the model pre-trained on PX, the scheme Base.+ CDR+ RCL ranks the first with performance degradation of 9.29%/15.18% in R-1/mAP on PX at the end of the 3rd stage, and with performance degradation of 24.44%/34.98% in R-1/mAP on PX at the end of the 6th stage. Besides, comparing the results in Tab.2 to the timely measured results in Tab.1, we find that the scheme Base.+ CDR+ RCL is of the lowest performance degradation over all domains. It also shows consistent trend with that in the Sec.5.2 of our manuscript, demonstrating the effectiveness of our proposed CLUDA-ReID in anti-forgetting. The above experiment results present that our proposed CLUDA-ReID performs consistently with different domain orders in the dynamic target scenario. This shows the practicality of CLUDA-ReID.

6. Ethical Impact

Our method is proposed to address more practical scenarios by achieving continuous domain adaptation using unlabeled streaming data in deployment environments. Our proposed task LUDA person ReID and its corresponding scheme CLUDA-ReID take an important next step in privacy protection since we do not need any identity annotations of real persons. (Note that we pre-train the person ReID models on synthetic data.) Despite this, it may still cause a violation of human privacy. Therefore, governments and officials need to carefully formulate strict regulations and laws to ensure the legal use of person ReID related technologies and strictly protect the data.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MS (t=1) R-1 mAP</th>
<th>MA (t=2) R-1 mAP</th>
<th>SY (t=3) R-1 mAP</th>
<th>MS (t=4) R-1 mAP</th>
<th>MA (t=5) R-1 mAP</th>
<th>SY (t=6) R-1 mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage-wise UDA (Base.)</td>
<td>24.16 9.13</td>
<td>66.69 40.99</td>
<td>57.83 54.42</td>
<td>23.43 8.90</td>
<td>68.92 43.43</td>
<td>56.33 52.83</td>
</tr>
<tr>
<td>Base. + Data Replay</td>
<td>26.08 9.45</td>
<td>68.08 43.26</td>
<td>59.14 59.03</td>
<td>27.38 9.83</td>
<td>70.34 47.34</td>
<td>61.43 59.24</td>
</tr>
<tr>
<td>Base. + CDR</td>
<td>28.03 11.03</td>
<td>70.32 46.42</td>
<td>63.12 64.34</td>
<td>29.34 11.63</td>
<td>72.43 52.31</td>
<td>67.21 67.82</td>
</tr>
<tr>
<td>Base. + CDR + KL Reg.</td>
<td>29.64 11.82</td>
<td>72.80 48.63</td>
<td>69.11 68.48</td>
<td>32.43 12.88</td>
<td>78.36 56.73</td>
<td>74.43 73.33</td>
</tr>
<tr>
<td>Base. + CDR + RCL</td>
<td>31.66 12.15</td>
<td>74.44 51.55</td>
<td>75.90 74.56</td>
<td>35.48 14.98</td>
<td>81.32 61.33</td>
<td>84.23 82.13</td>
</tr>
<tr>
<td>pretrain (t=0)</td>
<td>18.02 5.02</td>
<td>54.04 27.67</td>
<td>46.52 48.62</td>
<td>18.02 5.02</td>
<td>54.04 27.67</td>
<td>46.52 48.62</td>
</tr>
<tr>
<td>All-in-one UDA</td>
<td>36.37 17.84</td>
<td>85.12 68.59</td>
<td>86.76 84.70</td>
<td>36.37 17.84</td>
<td>85.12 68.59</td>
<td>86.76 84.70</td>
</tr>
</tbody>
</table>

Table 1. Adaptation performance (%) in the dynamic target scenario. We test the model instantly at the end of each stage, on its corresponding test set. The training order is PX → MS → MA → SY → MS → MA → SY. Our CLUDA-ReID is marked in gray shading.

<table>
<thead>
<tr>
<th>Methods</th>
<th>PX R-1 mAP</th>
<th>MS R-1 mAP</th>
<th>SY R-1 mAP</th>
<th>MA R-1 mAP</th>
<th>PX R-1 mAP</th>
<th>MS R-1 mAP</th>
<th>SY R-1 mAP</th>
<th>MA R-1 mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage-wise UDA (Base.)</td>
<td>48.95 27.13</td>
<td>10.83 3.01</td>
<td>50.98 25.22</td>
<td>40.83 16.45</td>
<td>8.83 2.79</td>
<td>48.42 21.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base. + Data Replay</td>
<td>58.26 39.29</td>
<td>18.60 6.01</td>
<td>56.15 29.41</td>
<td>52.39 24.28</td>
<td>20.11 7.83</td>
<td>58.32 31.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base. + CDR</td>
<td>63.93 44.33</td>
<td>20.49 8.21</td>
<td>59.74 32.70</td>
<td>58.46 39.64</td>
<td>22.21 9.03</td>
<td>63.74 35.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base. + CDR + KL Reg.</td>
<td>72.75 54.24</td>
<td>22.09 8.94</td>
<td>63.34 36.77</td>
<td>67.45 48.43</td>
<td>25.38 10.95</td>
<td>69.24 42.43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base. + CDR + RCL</td>
<td>85.57 71.16</td>
<td>26.16 11.07</td>
<td>69.75 43.70</td>
<td>70.42 51.36</td>
<td>31.46 13.91</td>
<td>77.34 50.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-trained Model (t=0)</td>
<td>94.86 86.34</td>
<td>18.02 5.02</td>
<td>54.04 27.67</td>
<td>94.86 86.34</td>
<td>18.02 5.02</td>
<td>54.04 27.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All-in-one UDA</td>
<td>40.43 16.44</td>
<td>36.37 17.84</td>
<td>85.12 68.59</td>
<td>40.43 16.44</td>
<td>36.37 17.84</td>
<td>85.12 68.59</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Anti-forgetting performance (%) evaluation in the dynamic target scenario. We test the model on the test-sets of all domains at the end of the 3rd stage (t=3) and the 6th stage (t=6). The training order is PX → MS → MA → SY → MS → MA → SY. Note that we omit the SY dataset for brevity because it appears in the final, whose results can not reflect the anti-forgetting ability. CLUDA-ReID is marked in gray.
References


