Weakly Supervised Open-set Domain Adaptation by Dual-domain Collaboration
Supplementary Material

Shuhan Tan$^{1,3,4}$, Jiening Jiao$^{2,3}$, Wei-Shi Zheng$^{1,3*}$

$^1$School of Data and Computer Science, Sun Yat-sen University, China
$^2$School of Electronics and Information Technology, Sun Yat-sen University, China
$^3$Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
$^4$Accuvision Technology Co. Ltd.

{tanshh, jiaojn}@mail2.sysu.edu.cn, wszheng@ieee.org

1. Optimization Method

We define the total loss function as

\[ f = \text{Dist}_C + \lambda_M \text{Dist}_M + \lambda_G G + \lambda_U U. \]  

(1)

Then, our objective function can be concluded as follows:

\[
(W_A^*, W_B^*) = \arg \min_{W_A, W_B} f(D_A, D_B, W_A, W_B).
\]  

(2)

The loss function \( f \) is not convex in the elements of \( W_A \) and \( W_B \). Therefore we use the non-linear methods in [23] to optimize the transform matrices.

We jointly optimize \( W_A \) and \( W_B \) by concatenating them as a whole matrix \( W \in \mathbb{R}^{d \times 2d} \). By optimizing \( W \) as a whole, we can ensure that both matrices are simultaneously updated. As for gradient \( \partial f/\partial W \), we similarly concatenate \( \partial f/\partial W_A \) and \( \partial f/\partial W_B \) to form a single gradient value. After the optimization of \( W_A \) and \( W_B \), we can easily obtain the projected features of \( D_A \) and \( D_B \) from \( D_A W_A \) and \( D_B W_B \).

To save space, we only give the gradient of each component in \( f \) for \( W_A \). The gradient of \( W_B \) can be similarly formed. From Equation

\[ f = \text{Dist}_C + \lambda_M \text{Dist}_M + \lambda_G G + \lambda_U U \]  

(3)

we have

\[
\frac{\partial f}{\partial W_A} = \frac{\partial \text{Dist}_C}{\partial W_A} + \lambda_M \frac{\partial \text{Dist}_M}{\partial W_A} + \lambda_G \frac{\partial G}{\partial W_A} + \lambda_U \frac{\partial U}{\partial W_A} \]  

(4)

For the first three components, the gradient can be simply formulated as

\[
\frac{\partial \text{Dist}_C}{\partial W_A} = \sum_{c \in C_K} (P_A^c)^T (P_A^c W_A - P_B^c W_B),
\]  

(5)

\[
\frac{\partial \text{Dist}_M}{\partial W_A} = (P_A)^T (P_A W_A - P_B W_B),
\]  

(6)

\[
\frac{\partial G}{\partial W_A} = \sum_{c \in C_L} \frac{1}{n_A^c} \sum_{x' \in X_A^c} (x' - \pi^c_A)^T (x' - \pi^c_A) W_A.
\]  

(7)

As for \( U \), the gradient is composed of two parts

\[
\frac{\partial U}{\partial W_A} = \frac{\partial U_A}{\partial W_A} + \frac{\partial U_B}{\partial W_A}
\]  

(8)

We first find indexes \((c, i) \in \mathcal{N}\) that triggers \( U_A \), then,

\[
\frac{\partial U_A}{\partial W_A} = \sum_{(c, i) \in \mathcal{N}} (x_i)^T (x_u W_{D'} - \pi^c_A W_A),
\]  

(9)

where \( D' \) denotes the host domain of \( x_u \). Then, we find the indexes \((c', i') \in \mathcal{N}\) that triggers \( U_B \), where the nearest unknown neighbor sample \( x_u \) is from \( D_A \). Therefore, the gradient for \( U_B \) can be defined as:

\[
\frac{\partial U_B}{\partial W_A} = \sum_{(c', i') \in \mathcal{N}} (x_u)^T (x_u W_A - x_i' W_B),
\]  

(10)

2. Time Complexity

We empirically compared the complexity of our method with top competitors on Intel E5-2650 CPU, with Nvidia GTX TITAN X GPU for the deep method. The time cost of each methods on different tasks are shown in Table 1. As shown, time complexity of CDA is comparable to other domain adaptation methods, while its accuracy outperformed all the compared methods.

3. Effect of Pseudo Labelling

We have compared this process with OSVM [9] and 1-vs-Set SVM [20]. We show the results for the first iteration of \( A \leftrightarrow W \) in Table 3, including accuracies of different
kinds of samples, rates of unknown-class samples mislabeled as known-class and vice versa. It is shown that both OSVM and 1-vs-Set SVM make imbalanced predictions by labelling too many known samples as unknown (90.91% and 95.33%), while our method has much better overall performances.

5. Effect of Overlapping Rate

In previous experiments, we set the overlapping rate of known samples ranging from 1 to 9. The result on the three Office tasks are shown in Fig. 1. The result showed that the performance of CDA gradually improves when more labeled samples are provided. This is expected because CDA use these labeled samples as initial inputs to assign pseudo labels for the unlabeled samples, and larger amount of labeled samples can produce more robust pseudo label prediction.

6. Additional Compared Methods

Besides the methods compared in the paper, we have also compared with other domain adaptation methods. We first compare with 3 traditional methods: JDA [16], SA [5] and JGSA [26]. Then, 4 Deep Learning methods are compared as well, including: RevGrad [6], DAN [14], DCORAL [22] and DDC [24].

The full results of these methods as well as the methods included in the paper are shown in Table 4 and Table 5, for Office and DukeMTMC-reID respectively.
<table>
<thead>
<tr>
<th>Method</th>
<th>r=5</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>AVG.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoAvg</td>
<td>45.0</td>
<td>53.3</td>
<td>49.0</td>
<td>54.2</td>
<td>59.2</td>
<td>98.9</td>
<td>40.0</td>
<td>95.4</td>
</tr>
<tr>
<td>NA</td>
<td>59.3</td>
<td>75.1</td>
<td>41.4</td>
<td>53.2</td>
<td>64.4</td>
<td>78.7</td>
<td>74.1</td>
<td>98.9</td>
</tr>
<tr>
<td>TCA [18]</td>
<td>38.6</td>
<td>74.3</td>
<td>41.0</td>
<td>53.7</td>
<td>64.3</td>
<td>78.0</td>
<td>75.3</td>
<td>98.7</td>
</tr>
<tr>
<td>GFK [7]</td>
<td>59.1</td>
<td>75.1</td>
<td>41.5</td>
<td>53.5</td>
<td>64.5</td>
<td>79.0</td>
<td>75.3</td>
<td>99.0</td>
</tr>
<tr>
<td>CORAL [11]</td>
<td>58.9</td>
<td>75.8</td>
<td>41.6</td>
<td>54.2</td>
<td>64.2</td>
<td>78.4</td>
<td>74.5</td>
<td>98.9</td>
</tr>
<tr>
<td>JAN [17]</td>
<td>24.0</td>
<td>43.7</td>
<td>34.4</td>
<td>64.4</td>
<td>21.4</td>
<td>38.6</td>
<td>75.5</td>
<td>90.2</td>
</tr>
<tr>
<td>PADA [5]</td>
<td>14.0</td>
<td>30.7</td>
<td>39.4</td>
<td>62.2</td>
<td>22.1</td>
<td>35.3</td>
<td>75.4</td>
<td>89.0</td>
</tr>
<tr>
<td>ATI [1]</td>
<td>58.6</td>
<td>74.0</td>
<td>40.6</td>
<td>49.5</td>
<td>65.4</td>
<td>79.2</td>
<td>36.1</td>
<td>78.0</td>
</tr>
<tr>
<td>OpenBIP [17]</td>
<td>34.7</td>
<td>49.7</td>
<td>45.9</td>
<td>66.5</td>
<td>27.3</td>
<td>39.8</td>
<td>77.1</td>
<td>90.6</td>
</tr>
<tr>
<td>JDA [16]</td>
<td>28.9</td>
<td>71.3</td>
<td>59.0</td>
<td>88.0</td>
<td>50.8</td>
<td>79.2</td>
<td>46.1</td>
<td>84.0</td>
</tr>
<tr>
<td>SA [5]</td>
<td>99.3</td>
<td>75.1</td>
<td>41.4</td>
<td>53.2</td>
<td>64.4</td>
<td>78.8</td>
<td>74.1</td>
<td>98.9</td>
</tr>
<tr>
<td>JGSA [16]</td>
<td>45.7</td>
<td>61.7</td>
<td>35.1</td>
<td>48.0</td>
<td>62.7</td>
<td>74.7</td>
<td>16.9</td>
<td>88.7</td>
</tr>
<tr>
<td>RevGrad [8]</td>
<td>53.6</td>
<td>71.9</td>
<td>35.9</td>
<td>50.5</td>
<td>54.6</td>
<td>68.8</td>
<td>67.5</td>
<td>97.4</td>
</tr>
<tr>
<td>DAN [14]</td>
<td>53.2</td>
<td>71.0</td>
<td>35.0</td>
<td>47.6</td>
<td>54.9</td>
<td>68.9</td>
<td>77.8</td>
<td>97.2</td>
</tr>
<tr>
<td>DCORAL [12]</td>
<td>52.2</td>
<td>70.8</td>
<td>36.4</td>
<td>50.3</td>
<td>55.0</td>
<td>69.1</td>
<td>78.5</td>
<td>97.4</td>
</tr>
<tr>
<td>MRN [13]</td>
<td>53.1</td>
<td>72.4</td>
<td>37.5</td>
<td>52.7</td>
<td>55.4</td>
<td>71.3</td>
<td>74.2</td>
<td>97.7</td>
</tr>
<tr>
<td>MMN [29]</td>
<td>36.0</td>
<td>51.1</td>
<td>35.7</td>
<td>51.5</td>
<td>57.5</td>
<td>73.2</td>
<td>74.2</td>
<td>98.9</td>
</tr>
<tr>
<td>LMNN [29]</td>
<td>59.4</td>
<td>78.1</td>
<td>42.5</td>
<td>59.1</td>
<td>61.8</td>
<td>74.1</td>
<td>84.7</td>
<td>96.3</td>
</tr>
<tr>
<td>KISSME [11]</td>
<td>55.2</td>
<td>68.6</td>
<td>49.2</td>
<td>71.0</td>
<td>63.1</td>
<td>75.3</td>
<td>86.9</td>
<td>94.9</td>
</tr>
<tr>
<td>XQDA [13]</td>
<td>55.6</td>
<td>68.9</td>
<td>49.1</td>
<td>71.0</td>
<td>63.1</td>
<td>75.3</td>
<td>86.0</td>
<td>94.9</td>
</tr>
<tr>
<td>DLLR [10]</td>
<td>59.2</td>
<td>75.0</td>
<td>41.3</td>
<td>52.9</td>
<td>64.1</td>
<td>78.1</td>
<td>72.9</td>
<td>98.7</td>
</tr>
<tr>
<td>SPGAN [4]</td>
<td>41.6</td>
<td>59.9</td>
<td>31.3</td>
<td>47.5</td>
<td>44.7</td>
<td>58.6</td>
<td>69.6</td>
<td>97.7</td>
</tr>
<tr>
<td>CDFA</td>
<td>64.6</td>
<td>80.4</td>
<td>62.6</td>
<td>88.1</td>
<td>60.9</td>
<td>84.9</td>
<td>76.0</td>
<td>99.9</td>
</tr>
</tbody>
</table>

Table 5. Comparing CMC accuracies with state-of-the-art methods on DukeMTMC-reID(%).
References


