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

$$f = Dist_C + \lambda_M Dist_M + \lambda_G G + \lambda_U U. \tag{1}$$

Then, our objective function can be concluded as follows:

$$(W_A^*, W_B^*) = \underset{W_A, W_B}{\arg\min} f(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}, W_A, W_B). \tag{2}$$

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 $\mathcal{D}_\mathcal{A}$ and $\mathcal{D}_\mathcal{B}$ from $\mathcal{D}_\mathcal{A} W_A$ and $\mathcal{D}_\mathcal{B} W_B$.

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

$$f = Dist_C + \lambda_M Dist_M + \lambda_G G + \lambda_U U \tag{3}$$

we have

$$\frac{\partial f}{\partial W_A} = \frac{\partial Dist_C}{\partial W_A} + \lambda_M \frac{\partial Dist_M}{\partial W_A} + \lambda_G \frac{\partial G}{\partial W_A} + \lambda_U \frac{\partial U}{\partial W_A} \tag{4}$$

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

$$\frac{\partial Dist_C}{\partial W_A} = \sum_{c\in\mathcal{C}_K} (P_A^c)^T (P_A^c W_A - P_B^c W_B), \tag{5}$$

$$\frac{\partial Dist_M}{\partial W_A} = (P_A)^T (P_A W_A - P_B W_B), \tag{6}$$

$$\frac{\partial G}{\partial W_A} = \sum_{c\in\mathcal{C}_\mathcal{L}} \frac{1}{n_A^c} \sum_{x^i\in\mathbf{X}_A^c} (x^i - \overline{x}_A^c)^T (x^i - \overline{x}_A^c) W_A. \tag{7}$$

As for U, the gradient is composed of two parts

$$\frac{\partial U}{\partial W_A} = \frac{\partial U_A}{\partial W_A} + \frac{\partial U_B}{\partial W_A} \tag{8}$$

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

$$\frac{\partial U_A}{\partial W_A} = \sum_{(c,i)\in\mathcal{N}} (x^i)^T (x_u W_{D'} - \overline{x}_A^c W_A), \tag{9}$$

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 $\mathcal{D}_\mathcal{A}$. Therefore, the gradient for U_B can be defined as:

$$\frac{\partial U_B}{\partial W_A} = \sum_{(c',i')\in\mathcal{N}} (x_u)^T (x_u W_A - x^{i'} W_B). \tag{10}$$

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 ↔ W in Table 3, including accuracies of different

Methods	A↔D	A↔W	W↔D	5↔6	7↔8
MMDT [8]	10.23	9.56	9.00	368.80	414.72
TCA [18]	2.04	1.92	0.59	25.41	39.04
OpenBP [19]	16109.38	11120.06	6061.15	6445.76	12559.27
CDA	7.28	8.03	4.72	112.05	169.10

Table 1. Time complexity of different methods (second).

Overlapping rate	0%	25%	50%	75%	100 %
TCA	71.1	70.1	73.3	65.4	65.2
ATI-semi	73.8	72.5	**73.4**	**73.8**	**74.8**
MMDT	**75.5**	**75.8**	72.6	73.5	74.6
CDA	**76.8**	**77.8**	**77.1**	**77.6**	**80.0**

Table 3. Impact of the overlapping proportion(%).

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.

Method	overall	known	unknown	u → k	k → u
OSVM	58.12	9.09	**99.89**	**0.11**	90.91
1-vs-Set SVM	55.93	4.67	**99.84**	**0.16**	95.33
LSVM w/o outlier detection	**69.19**	**63.41**	74.12	25.88	**17.80**
LSVM w/ outlier detection (ours)	**85.64**	**81.57**	89.10	10.90	**10.11**

Table 2. The effect of pseudo-labeling process(%).

4. Effect of Number of Labeled Samples

In *Office* experiment, we set the number of labeled samples added from each labeled known categories to 3. We also analysis the impact of this number of labeled samples per class by conducting CDA with the number of labeled samples ranging from 1 to 9. The result on the three *Office* tasks are shown in Fig. 1.

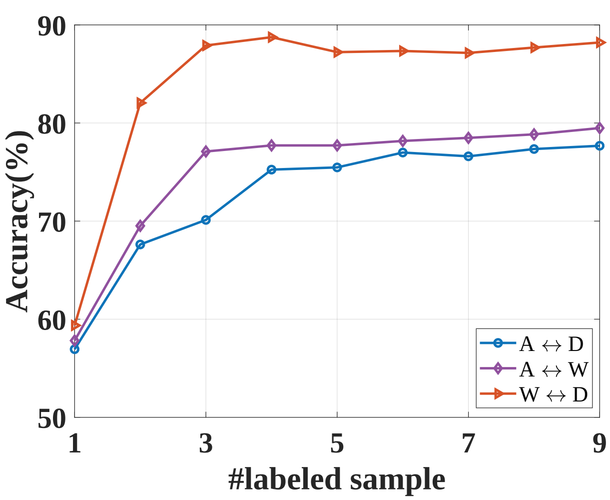

Figure 1. Impact of labeled sample number per class

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.

5. Effect of Overlapping Rate

In previous experiments, we set the overlapping rate of known label spaces between the two domains to 50% (*e.g.*, class 1-10 for $\mathcal{D}_{\mathcal{A}}$, 6-15 for $\mathcal{D}_{\mathcal{B}}$). To compare the performance of our method under different overlapping rates, we have conducted experiments with overlapping rate varying from 0 to 1. We show the results of top competitors on the *Office* task A ↔ W in Table 3, and CDA performed the best under different circumstances.

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.

Methods	A ↔ W	A ↔ D	W ↔ D	AVG.
NA-avg	62.2 ± 3.19	61.0 ± 3.05	66.3 ± 1.72	63.12
NA	72.6 ± 2.04	69.4 ± 2.22	84.2 ± 3.89	75.40
TCA [18]	73.3 ± 1.67	70.6 ± 1.99	83.5 ± 3.96	**75.80**
GFK [7]	56.9 ± 2.89	55.7 ± 1.46	70.9 ± 4.85	61.17
CORAL [21]	69.9 ± 4.41	67.7 ± 3.13	83.7 ± 3.65	73.77
JAN [17]	63.8 ± 1.27	65.5 ± 0.76	74.7 ± 1.41	68.00
PADA [2]	60.3 ± 1.09	60.2 ± 0.98	70.9 ± 1.88	63.80
ATI [1]	70.4 ± 4.15	65.9 ± 1.80	81.7 ± 4.74	72.67
OpenBP [19]	62.6 ± 4.11	62.9 ± 1.71	67.9 ± 2.31	64.47
JDA [16]	72.6 ± 2.21	70.0 ± 2.03	83.2 ± 4.19	75.27
SA [5]	72.2 ± 2.07	69.9 ± 2.18	82.6 ± 3.87	74.90
JGSA [26]	62.3 ± 1.97	62.5 ± 1.05	67.2 ± 4.14	64.00
RevGrad [6]	57.1 ± 1.75	56.9 ± 1.60	58.0 ± 1.01	57.33
DDC [24]	55.1 ± 0.82	55.2 ± 0.94	55.6 ± 0.93	55.30
DAN [14]	69.4 ± 2.42	66.5 ± 4.13	83.6 ± 3.22	73.17
DCORAL[22]	68.6 ± 1.50	67.6 ± 4.02	83.5 ± 2.71	73.23
ATI-semi [1]	**73.4 ± 2.28**	**72.0 ± 2.87**	77.8 ± 3.40	74.72
MMDT [8]	72.6 ± 2.15	69.4 ± 2.19	**84.4 ± 3.99**	75.47
AMTL [12]	50.2 ± 1.45	48.8 ± 0.90	62.1 ± 2.17	53.70
CLMT [3]	50.3 ± 1.47	50.0 ± 0.77	61.7 ± 1.23	54.00
MRN [15]	62.2 ± 3.38	62.4 ± 2.44	77.4 ± 3.43	67.33
CDA	**77.1 ± 1.35**	**75.2 ± 1.63**	**88.1 ± 2.45**	**80.13**

Table 4. Comparing state-of-the-art methods on *Office*. The 1st/2nd best results are indicated in red/blue.

Method	1 ↔ 2		2 ↔ 3		3 ↔ 4		4 ↔ 5		5 ↔ 6		6 ↔ 7		7 ↔ 8		8 ↔ 1		AVG.	
	r=1	r=5	r=1	r=5	r=1	r=5	r=1	r=5	r=1	r=5	r=1	r=5	r=1	r=5	r=1	r=5	r=1	r=5
NA-avg	43.6	55.3	29.6	38.2	50.1	54.8	69.2	95.2	38.2	54.4	33.5	42.4	22.0	40.3	76.7	95.4	45.4	59.5
NA	59.3	75.1	41.4	53.2	64.4	78.8	74.1	98.9	49.0	67.1	51.9	65.4	27.6	49.6	78.3	98.9	55.7	73.4
TCA [18]	58.6	74.7	41.0	53.7	64.3	79.0	75.3	98.9	48.6	67.3	51.5	65.3	27.2	49.8	78.3	98.9	55.6	73.5
GFK [7]	59.1	75.1	41.5	53.5	64.5	79.0	75.3	**99.0**	49.0	67.4	51.9	65.5	27.7	50.3	78.5	98.9	55.9	73.6
CORAL [21]	58.9	75.8	41.6	54.2	64.2	78.4	74.5	98.8	47.7	66.3	51.5	64.6	26.5	48.6	78.4	98.8	55.4	73.2
JAN [17]	24.0	43.7	34.4	64.4	21.4	38.6	75.5	90.2	30.2	60.0	27.7	52.4	44.6	72.1	81.7	92.5	42.4	64.2
PADA [2]	14.0	30.7	39.4	62.2	22.1	35.3	75.4	89.0	30.1	57.4	27.2	50.4	47.4	70.9	77.6	91.1	41.6	60.9
ATI [1]	58.6	74.0	40.6	49.5	65.4	79.2	36.1	78.0	47.9	63.6	52.4	65.6	24.6	42.7	28.2	84.4	44.2	67.1
OpenBP [19]	34.7	49.7	45.9	66.5	27.3	39.8	77.1	90.6	8.1	26.5	37.6	53.5	47.5	67.3	83.5	93.3	45.2	54.9
JDA [16]	58.9	73.4	39.6	48.6	**66.9**	**79.5**	43.7	84.0	45.2	59.1	49.7	60.7	22.8	36.0	35.9	84.2	45.3	65.7
SA [5]	59.3	75.1	41.4	53.2	64.4	78.8	74.1	98.9	49.0	67.1	51.9	65.4	25.7	49.7	78.3	**99.0**	55.5	73.4
JGSA [26]	45.7	61.7	35.1	48.0	62.7	74.7	16.9	88.7	40.3	53.5	43.2	55.6	16.5	31.3	7.2	36.5	33.5	56.2
RevGrad [6]	53.6	71.9	35.9	50.5	54.6	68.8	67.5	97.4	47.8	68.5	45.1	65.7	31.0	60.6	87.6	96.3	52.9	72.5
DAN [14]	53.2	71.0	35.0	47.6	54.9	68.9	77.8	97.2	45.5	66.3	45.6	65.2	31.3	59.9	82.6	97.2	53.2	71.7
DCORAL [22]	52.2	70.8	36.4	50.3	55.0	69.1	78.5	97.4	48.8	70.3	46.7	66.2	31.8	61.2	**89.2**	96.5	54.8	72.7
DDC [24]	53.1	72.4	37.5	52.7	55.4	71.3	74.2	97.7	46.4	66.7	45.2	63.8	30.6	58.3	**92.3**	95.6	54.3	72.3
MRN [15]	26.2	46.5	19.2	32.8	34.0	49.6	74.9	94.4	25.9	49.9	24.4	43.1	19.9	50.2	53.6	85.1	34.8	56.5
ATI-semi [1]	53.4	71.8	**55.6**	**71.8**	56.5	76.8	39.7	88.9	55.3	69.3	**57.6**	**76.9**	**46.7**	65.6	30.6	86.3	49.4	75.9
MMDT [8]	36.0	51.1	35.7	51.5	57.5	73.2	74.2	98.9	29.2	53.2	22.0	39.5	19.1	43.5	76.5	98.6	42.9	62.3
LMNN [25]	**59.4**	**78.1**	42.5	59.1	61.8	74.1	84.7	96.3	53.3	72.2	50.8	66.1	33.5	60.8	90.2	97.7	**59.5**	75.5
KISSME [11]	55.2	68.6	49.2	71.0	63.1	75.3	**86.0**	94.9	**57.6**	74.5	55.3	71.6	45.8	**71.6**	22.0	56.7	54.3	73.0
XQDA [13]	55.6	68.9	49.1	71.0	63.1	75.3	**86.0**	94.9	57.6	**74.5**	55.3	71.6	45.8	71.6	48.2	86.0	57.6	**76.7**
DLLR [10]	59.2	75.0	41.3	52.9	64.1	78.1	72.9	98.7	48.7	66.9	51.8	64.5	27.9	49.6	78.5	**99.0**	55.6	73.1
SPGAN [4]	41.6	59.9	31.3	47.5	44.7	58.6	69.6	97.7	41.4	61.0	43.3	59.6	22.8	46.3	74.1	98.0	48.7	69.6
CDA	**64.6**	**80.4**	**62.4**	**88.1**	**67.9**	**84.9**	76.0	**98.9**	**61.5**	**82.1**	**59.6**	**78.1**	**67.2**	**83.8**	85.6	98.5	**68.1**	**86.9**

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

References

[1] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 754–763, 2017. 2, 3

[2] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018. 2, 3

[3] Carlo Ciliberto, Youssef Mroueh, Tomaso Poggio, and Lorenzo Rosasco. Convex learning of multiple tasks and their structure. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1548–1557, 2015. 2

[4] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[5] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2960–2967, 2013. 2, 3

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015. 2, 3

[7] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012. 2, 3

[8] Judy Hoffman, Erik Rodner, Jeff Donahue, Kate Saenko, and Trevor Darrell. Efficient learning of domain-invariant image representations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2013. 2, 3

[9] Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. Multi-class open set recognition using probability of inclusion. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 393–409, Cham, 2014. Springer International Publishing. 1

[10] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Dictionary learning with iterative laplacian regularisation for unsupervised person re-identification. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 3, page 8, 2015. 3

[11] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2295. IEEE, 2012. 3

[12] Giwoong Lee, Eunho Yang, et al. Asymmetric multi-task learning based on task relatedness and confidence. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 230–238, 2016. 2

[13] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2197–2206, 2015. 3

[14] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 97–105, 2015. 2, 3

[15] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and S Yu Philip. Learning multiple tasks with multilinear relationship networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 1594–1603, 2017. 2, 3

[16] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jia-Guang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2200–2207, 2013. 2, 3

[17] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2208–2217, 2017. 2, 3

[18] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011. 2, 3

[19] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[20] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, July 2013. 1

[21] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the*

Thirtieth AAAI Conference on Artificial Intelligence, AAAI, pages 2058–2065. AAAI Press, 2016. 2, 3

[22] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 443–450. Springer, 2016. 2, 3

[23] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM journal on optimization*, 12(2):555–573, 2002. 1

[24] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR, abs/1412.3474*, 2014. 2, 3

[25] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009. 3

[26] Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5150–5158. IEEE, 2017. 2, 3