Supplementary Material for
Generalized Domain Adaptation

Yu Mitsuzumi Go Irie Daiki Ikami Takashi Shibata
NTT Communication Science Laboratories, NTT Corporation, Japan
{yu.mitsuzumi.ae, daiki.ikami.ef}@hco.ntt.co.jp, {goirie, t.shibata}@ieee.org

Abstract

In this supplementary material for Generalized Domain Adaptation (main paper), we provide details of the network architectures used in our experiments and present additional experimental results.

A. Network Architectures

This section provides the details of the network architectures used in our experiments (Sec. 5 in the main paper). We consistently use the same network architecture detailed in Table 1 for our domain label estimation throughout all the experiments in the main paper. The architecture of our classifier network for Digits is shown in Table 2.

Table 1: **Network architecture for domain label estimation**.

$32 \times 32 \times 1$ (Digits) or $32 \times 32 \times 3$ (Office-31/Office-Home) Input
3×3 conv. 16 channels, ReLU, BatchNorm
3×3 conv. 32 channels, ReLU, BatchNorm
2×2 Max Pooling
3×3 conv. 64 channels, ReLU, BatchNorm
3×3 conv. 64 channels, ReLU, BatchNorm
2×2 Max Pooling
Average Pooling
Fully Connected Layer 64, ReLU, BatchNorm
Fully Connected Layer 64

Table 2: **Network architecture of class classifier for Digits.**

Feature Extractor (G_f)	
$32 \times 32 \times 3$ Input	
InstanceNorm	
5×5 conv. 64 channels, LeakyReLU, SwitchNorm [10]	
5×5 conv. 64 channels, LeakyReLU, SwitchNorm [10]	
3×3 conv. 128 channels, stride 2, LeakyReLU, SwitchNorm [10]	
3×3 conv. 128 channels, stride 2, LeakyReLU, SwitchNorm [10]	
Dropout	
Class Label Predictor (F_y)	**Domain Classifier (F_d)**
	Gradient Reversal Layer
Fully Connected Layer 100, ReLU, SwitchNorm [10]	Fully Connected Layer 100, ReLU, SwitchNorm [10]
	Dropout
Fully Connected Layer 100, ReLU, SwitchNorm [10]	Fully Connected Layer 100, ReLU, SwitchNorm [10]
	Dropout
Fully Connected Layer #class Softmax	Fully Connected Layer #domain Softmax

B. Additional Results for Office-Home

We report additional results for Office-Home in the MS-OSDA and BTDA problems. We follow the protocols used in [12, 1], as we did in the experiments (Sec. 5.2) described in the main paper. For MS-OSDA, we consider all four possible combinations of three source domains and one target domain with 45 shared and 20 open set classes. For BTDA, we consider all four possible combinations of one source domain and three target domains.

Results for MS-OSDA. Table 3 shows the results. Our method is the second best and outperforms MOS-DANET [12], the state-of-the-art MS-OSDA method, in one condition.

Results for BTDA. Table 4 shows the results. Our method is competitive with AMEAN [1] and consistently better than the other methods.

C. Comparison with OCDA

Liu et al. [6] recently proposed a new UDA variant called open compound domain adaptation (OCDA). The OCDA problem deals with adaptation to "open domains," i.e., domains not present in the training data, and they proposed a solution for the problem. Although the OCDA problem is not within the scope of our GDA, evaluating the performance of the OCDA method on our GDA problem and the performance of our GDA method on the OCDA problem would be interesting.

Evaluation of OCDA Method in GDA Problem. We first show the performance of the OCDA method [6] in our GDA1 problem. The protocol is exactly the same as the one mentioned in the main paper. Note that we use only the OS*

Table 3: **Results for Office-Home in MS-OSDA**. OS values are listed in the table. The best and second-best are highlighted in bold and underlined, respectively.

	Ar, Cl, Pr → Rw	Ar, Pr, Rw → Cl	Pr, Cl, Rw → Ar	Ar, Cl, Rr → Pr	Avg.
OSVM [14]	60.2	46.3	48.6	57.0	53.0
OSVM+DANN [3]	54.5	31.6	40.9	53.8	45.2
OSBP [13]	53.6	38.0	46.9	54.9	48.3
IOSBP [2]	64.5	46.2	54.9	66.4	58.1
MOSDANET [12]	**80.3**	**67.5**	60.6	**80.0**	**72.1**
Ours	78.6	59.1	**64.8**	75.5	69.5

Table 4: **Results for Office-Home in BTDA**. Classification accuracy is shown in the table. The best and second-best are highlighted in bold and underlined, respectively.

	Ar → Cl, Pr, Rw	Cl → Ar, Pr, Rw	Pr → Ar, Cl, Rw	Rw → Ar, Cl, Pr	Avg.
Labeled only	47.6	42.6	44.2	51.3	46.4
DAN [7]	55.6	56.6	48.5	56.7	54.4
RTN [8]	53.9	56.7	47.3	51.6	52.4
JAN [9]	58.3	60.5	52.2	57.5	57.1
RevGrad [3]	58.4	58.1	52.9	62.1	57.9
AMEAN [1]	**64.3**	**65.5**	**59.5**	**66.7**	**64.0**
Ours	63.3	63.6	58.6	64.8	62.6

metric because the OCDA method has no mechanism to detect unknown classes. Table 5 shows the results. Compared with the best baseline in GDA1, OSBP, the OCDA method won some and lost some. Ours is clearly better than the OCDA method in all the setups, which stresses the merit of our method.

Evaluation of GDA method in OCDA Problem. We next report the performance of our method in the OCDA problem. In the experiment, we use SVHN [11] as a source domain, MNIST [5], MNIST-M [3], and USPS [4] as compound domains, which consist of multiple target domains without their domain labels, and SynDigits [3] as an open domain. We compare our method with the OCDA method [6] and AMEAN [1]. The results are shown in Table 6. Note that the symbol ‡ means that open domain images are treated as being included in compound domains during training. While Ours is highly competitive with AMEAN, the best baseline used in [6], it cannot outperform the OCDA method [6]. Extending our method to a form applicable to the OCDA problem would be an interesting future direction.

D. Additional Analysis

Ablation of Regularization Term $\mathcal{L}_p$ [15]. The regularization term $\mathcal{L}_p$ [15] requires the true class distribution of the data in advance, which cannot be known in practice. We evaluated our method without $\mathcal{L}_p$ to ascertain the performance when the distribution is unknown. The HOS values on Office-31 were 81.02 with $\mathcal{L}_p$ vs. 80.34 without $\mathcal{L}_p$. The gap was only 0.7%, which proves the strong robustness of our method.

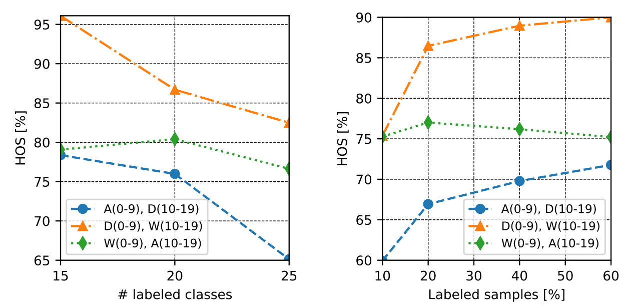
Performance to Different Difficulty Levels. The difficulty of the GDA task varies depending on the number of classes to be classified and the percentage of labeled data. We evaluate the performance of the proposed method under various difficulty levels of GDA. Fig. 1a shows the accuracy for different numbers of labeled classes on Office-31 in GDA1. As with general multiclass classification, the accuracy decreases as the number increases. Fig. 1b shows the accuracy for different ratios of labeled samples on Office-31 in GDA2. As the ratio decreases, the accuracy decreases.

Table 5: **OCDA results for Digits in GDA1.** OS* values are listed in the table. The best are highlighted in bold.

Setup	sv(0-3), sy(4-7)	sv(0-3), mt(4-7)	sv(0-2), sy(3-5), mt(6-8)
OSBP [13]	64.33	7.37	33.95
OCDA [6]	34.35	0.83	38.49
Ours	**86.18**	**70.50**	**79.76**
Setup	sv(0,1), sy(2,3), mt(4,5), mm(6,7)	sv(0-5), mt(2-7)	sv(0-5), mt(2-7)
OSBP [13]	19.48	33.15	10.93
OCDA [6]	17.08	57.22	4.58
Ours	**66.02**	**85.83**	**73.46**

Table 6: **Results for OCDA problem**. Classification accuracy is reported in the table. The best are highlighted in bold.

Source	Compound Domains			Open	
SVHN	MNIST	MNIST-M	USPS	SynDigits	Avg.
AMEAN‡ [1]	85.2	**65.7**	74.3	84.4	77.4
OCDA [6]	**90.9**	**65.7**	**83.4**	88.2	**82.1**
Ours‡	81.7	57.9	77.4	**92.3**	77.3
Ours	81.3	59.5	76.3	87.6	76.2



(a) (b)

Figure 1: **Performance of GDA with various difficulty levels.**

References

[1] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proc. CVPR*, 2019. 1, 2

[2] Jiahui Fu, Xiaofu Wu, Suofei Zhang, and Jun Yan. Improved open set domain adaptation with backpropagation. In *Proc. ICIP*, 2019. 2

[3] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. ICML*, 2015. 2

[4] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5):550–554, 1994. 2

[5] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. 2

[6] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X Yu, and Boqing Gong. Open compound domain adaptation. In *Proc. CVPR*, 2020. 1, 2

[7] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proc. ICML*, 2015. 2

[8] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proc. NeurIPS*, 2016. 2

[9] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proc. ICML*, 2017. 2

[10] Ping Luo, Jiamin Ren, Zhanglin Peng, Ruimao Zhang, and Jingyu Li. Differentiable learning-to-normalize via switchable normalization. In *Proc. ICLR*, 2019. 1

[11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Proc. NeurIPS*, 2011. 2

[12] Sayan Rakshit, Dipesh Tamboli, Pragati Shuddhodhan, Biplab Banerjee Meshram, Gemma Roig, and Subhasis Chaudhuri. Multi-source open-set deep adversarial domain adaptation. In *Proc. ECCV*, 2020. 1, 2

[13] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proc. ECCV*, 2018. 2

[14] Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. Probability models for open set recognition. *IEEE TPAMI*, 36(11):2317–2324, 2014. 2

[15] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proc. CVPR*, 2018. 2