Supplementary Material: Re-energizing Domain Discriminator with Sample Relabeling for Adversarial Domain Adaptation

Xin Jin[1] Cuiling Lan[2]* Wenjun Zeng[2] Zhibo Chen[1]*

[1] University of Science and Technology of China [2] Microsoft Research Asia, Beijing, China

jinxustc@mail.ustc.edu.cn {culan,wezeng}@microsoft.com chenzhibo@ustc.edu.cn

We provide more implementation details and experimental results in this supplementary.

1. More Training Details

In this section, we present more training and implementation details about our RADA and baseline setup on the different datasets. Following [3], we initialize the feature extractor backbone (*e.g.*, ResNet-50, ResNet-101) with the pre-trained parameters on ImageNet. We employ stochastic gradient descent (SGD) as optimizer with a momentum term of 0.9. For the data augmentation, we perform random flipping (horizontally), and scaling with a factor in 1.125 followed by a crop. More details are described as follows.

For *Office-31* and *Office-Home*, following [6, 16, 3], we use ResNet-50 as backbone. The initial learning rate is set to 1e-3. The input image size is 224×224 and the batch size is 36. We train the models for 100 epochs and evaluate their adaptation performance. We use the default train/test/val split protocol as [3, 6] for both the two datasets.

For *VisDA-2017*, following [6, 3], we use ResNet-50 as backbone. The initial learning rate is set to 1e-4. The input image size is 224×224, and the batch size is 36. We follow the train/val/test split protocol of [3] and train the models for 150 epochs.

For *Digit-Five*, following [12, 7, 13], we use Cov_3FC_2 as backbone which is trained from scratch. The initial learning rate is set to 0.05. The input image size is 32×32, and the batch size is 256 (256 = 4×64 where 4 means the number of source domains). We follow the train/val/test split protocol of [7]. We train the models for 50 epochs.

For *DomainNet*, following [7], we use ResNet-101 as backbone. The initial learning rate is set to 0.002 and momentum set to 0.9. The input image size is 224×224, and the batch size is 30 (30 = 5×6 where 5 means the number of source domains). We follow the train/val/test split protocol of [7]. We train the models for 40 epochs.

2. More Details about When to Start RADA

As we have described in the main manuscript, inspired by the popular learning rate (lr) adjustment algorithms [8, 1] which adjust the learning rate if no improvement is seen for a 'patience' number of epochs (where 'patience' is usually set to 2 to 10) [1], we start RADA if no improvement of the discrimination capability is seen for a 'patience' number of epochs and we denote this hyper-parameter as K. Note that once RADA is started, it will be used for each iteration. Particularly, as described in lines128-136 in our main manuscript, we use the average entropy e_i of domain classification for all the training samples to measure the discrimination capability of the domain discriminator in the i^{th} epoch. A larger average entropy indicates a poorer discrimination capability of the domain discriminator. RADA starts in the i^{th} epoch when no improvement of the discrimination capability (average entropy) is seen for K epochs, *i.e.*, $e_t \geq e_{i-K-1}$, for $t = i-1$ to $t = i-K$.

3. Adversarial Training on a Mini-batch After Relabeling

As is done in previous adversarial domain adaptation methods, we perform mini-batch level optimization where a batch consists of both source and target domain samples. Once RADA is activated, for each mini-batch, we first check whether each target sample should be relabeled as a source sample and relabel them if deemed so. Then the adversarial training is performed under the updated domain

*Corresponding Author.

[1] https://pytorch.org/docs/stable/optim.html?highlight=reducelronplateau #torch.optim.lr_scheduler.ReduceLROnPlateau

labels. The new domain adversarial loss is thus

$$\begin{aligned}\mathcal{L}_{adv}^{new} = &- \frac{1}{n_s + n_{t\to s} + n_m}\Big(\sum_{i=1}^{n_s} \log D(F(\mathbf{x}_i^s)) \\ &+ \sum_{j=1}^{n_{t\to s}} \log D(F(\mathbf{x}_j^{t\to s})) + \sum_{k=1}^{n_m} \log D(\widetilde{\mathbf{f}}_k^s)\Big) \\ &- \frac{1}{n_t - n_{t\to s}} \sum_{l=1}^{n_t - n_{t\to s}} \log(1 - D(F(\mathbf{x}_l^t))),\end{aligned}$$

where n_s, n_t, $n_{t\to s}$, n_m, $n_t - n_{t\to s}$ denote the number of original source samples $\mathbf{x}_i^s$, original target samples, relabeled target samples $\mathbf{x}_j^{t\to s}$, the mixed/generated source samples $\widetilde{\mathbf{f}}_k^s$ (see description around Eq. (6) in our manuscript), and the remaining target samples $\mathbf{x}_l^t$, in a minbatch, respectively. For simplicity, we set $n_m = n_{t\to s}$.

4. More Experimental Results

More Results Showing the Degradation of Domain Discriminator vs. Our Solution. In our main manuscript, we have revealed the degradation problem of the domain discriminator of the baseline scheme in Figure 2 (Figure 1 here), and showed that our scheme alleviates this and achieves a better alignment state (*i.e.*, a lower domain discrepancy measurement) on Office-31 (with the analysis presented in lines140-188 and lines195-209).

Similar trends have been observed on the other datasets. Figure 2 shows the results on Office-Home of the setting Ar→Cl, and VisDA-2017. For the baseline scheme CDAN [6] (marked by green), the discrimination capability of the domain discriminator deteriorates w.r.t. the gradually aligned distributions after the initial dip of the entropy, which in turn provides less driving power to the feature extractor for alignment. In contrast, thanks to our strategy of re-labeling the aligned target samples as source samples, our scheme (marked by red) could improve the discrimination capability of the domain discriminator (*i.e.*, preventing the increasing of the entropy) and thus in turn further drives feature alignment.

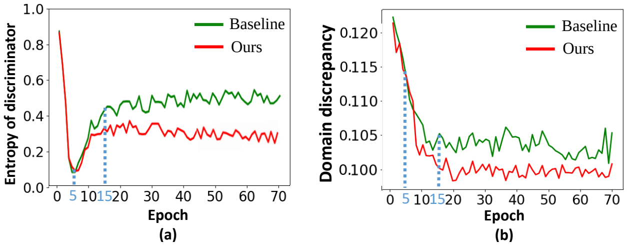
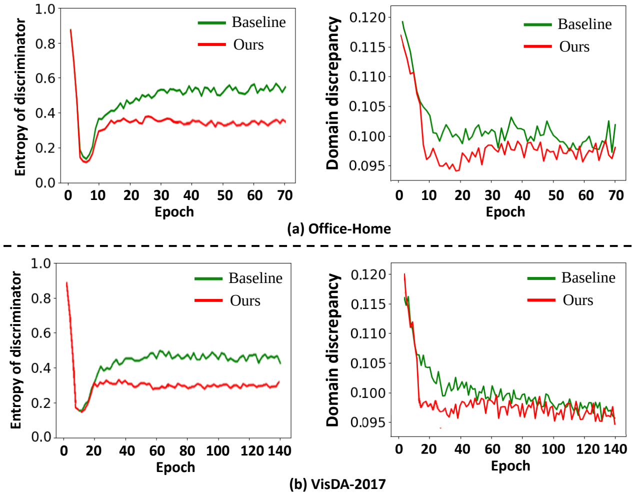


Figure 1: The variation of (a) the discrimination capability of the domain discriminator (measured by entropy of domain classification) and (b) alignment state (measured by domain discrepancy measure of MMD) in the training. These experiments are conducted on Office-31 of the setting W→A.



(a) Office-Home

(b) VisDA-2017

Figure 2: The variation of the discrimination capability of the domain discriminator (measured by entropy of domain classification) and alignment state (measured by domain discrepancy measure of MMD) in the training. These experiments are conducted on (a) Office-Home of the setting Ar→Cl and on (b) VisDA-2017.

Influence of Metric for Measurement of Alignment. In RADA, as described in our main manuscript, to measure whether a target sample is a "well aligned" sample or not, we propose to simply use the entropy of domain discriminator to make decision. When the entropy of the domain discriminator is larger than a threshold τ, we define it as a "well aligned" sample, where τ is a hyper-parameter (see ablation study in Figure 4 in our main manuscript).

Moreover, we have studied the influence of using different metric designs for the measurement. We compare several schemes. 1) $\mathcal{H}_C$: we use the entropy of the **object** classifier, *i.e.*, $\mathcal{H}(C(F(\cdot)))$, as metric to select "well-aligned" target samples. For a target sample, when the entropy of the object classification is smaller than a threshold (which indicates the sample is easy to transfer), we define it as a "well aligned" sample, where the threshold is a hyper-parameter obtained by grid search. p_{src}: For a target sample, we use the predicted probability of being source domain, *i.e.*, p_{src}, as metric to select "well-aligned" target samples. When p_{src} is larger than a threshold (which indicates the sample is near to the source domain), we define it as a "well aligned" sample, where the threshold is a hyper-parameter obtained by grid search. $\mathcal{H} * p_{src}$: we use the product of the entropy of the domain discriminator $\mathcal{H}$ and the predicted probability of being source domain p_{src}, as metric to select "well-aligned" target samples. When $\mathcal{H} * p_{src}$ is larger than a threshold, we define it as a "well aligned" sample, where the threshold is a hyper-parameter obtained by grid search. $\mathcal{H}$: our scheme which simply uses the entropy of the

Table 1: Ablation study on the different design choices for the metric to select "well aligned" target samples.

Method	VisDA-2017
DANN (Baseline) [4]	61.23
DANN + RADA w/ $\mathcal{H}_C$	63.78
DANN + RADA w/ p_{src}	64.51
DANN + RADA w/ $\mathcal{H} * p_{src}$	65.77
DANN + RADA w/ $\mathcal{H}$	**65.91**
CDAN (Baseline) [6]	70.82
CDAN + RADA w/ $\mathcal{H}_C$	72.87
CDAN + RADA w/ p_{src}	74.18
CDAN + RADA w/ $\mathcal{H} * p_{src}$	**75.83**
CDAN + RADA w/ $\mathcal{H}$	75.62

domain discriminator $\mathcal{H}$ as metric to select "well-aligned" target samples.

Table 1 shows the results. The schemes *CDAN+RADA w/ $\mathcal{H}$* and *CDAN+RADA w/ $\mathcal{H} * p_{src}$* achieve very similar performance but significantly outperform the baseline and our schemes using other metrics. We analyse that simultaneously considering the entropy of the domain discriminator $\mathcal{H}$ and the predicted probability of being source domain p_{src} should better cover the "well-aigned" target samples. Because when some well-aligned target samples are very close to the source domain center but far away from the boundary of the domain discriminator, these target samples would have small entropy value w.r.t. the domain discriminator, which should be relabeled but may be missed when only the entropy is taken as the metric. We found *CDAN+RADA w/ $\mathcal{H}$* and *CDAN+RADA w/ $\mathcal{H} * p_{src}$* achieve very similar performance, which may because the above cases are rare in practices. For simplicity, we use the entropy of the domain discriminator as the metric to select "well aligned" samples by default, *i.e.*, *CDAN+RADA w/ $\mathcal{H}$*.

5. Comparison with State-of-the-Arts (Complete Version)

For Table 4 (a)(c) and (d) in our main manuscript, to save space, we only present the average accuracy. Here, we also present the detailed results of each sub-setting in Table 2, Table 3, and Table 4, for Office-Home, Digit-Five, and DomainNet, respectively. We can observe that our *CDAN+RADA* achieves the state-of-the-art performance.

References

[1] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012. 1

[2] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, pages 3941–3950, 2020. 4

[3] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, pages 12455–12464, 2020. 1, 4

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 3, 4

[5] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*, 2020. 4

[6] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1640–1650, 2018. 1, 2, 3, 4

[7] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 1, 4

[8] Aaditya Prakash, James Storer, Dinei Florencio, and Cha Zhang. Repr: Improved training of convolutional filters. In *CVPR*, pages 10666–10675, 2019. 1

[9] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018. 4

[10] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, pages 8725–8735, 2020. 4

[11] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI*, volume 33, pages 5345–5352, 2019. 4

[12] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *CVPR*, pages 3964–3973, 2018. 1, 4

[13] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. *ECCV*, 2020. 1, 4

[14] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *ICML*, pages 7124–7133, 2019. 4

[15] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. *ICML*, 2019. 4

[16] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pages 5031–5040, 2019. 1, 4

[17] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *NeurIPS*, pages 8559–8570, 2018. 4

Table 2: Performance (%) comparisons on Office-Home with the state-of-the-art approaches for unsupervised domain adaptation. All experiments are based on ResNet-50 pre-trained on ImageNet.

Method	Venue	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
DANN [4]	JMLR'16	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
MCD [9]	CVPR'18	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [6]	NeurIPS'18	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MDD [15]	ICML'19	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
Symnets [16]	CVPR'19	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
TADA [11]	AAAI'19	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
BNM [2]	CVPR'20	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
Symnets-GVB [3]	CVPR'20	48.0	74.3	78.5	65.1	72.2	74.4	65.1	49.4	79.7	73.8	51.7	82.5	67.8
CDAN-GVB [3]	CVPR'20	55.3	74.1	78.2	62.4	72.6	71.8	63.8	54.1	80.1	73.1	58.7	83.6	69.0
GVB [3]	CVPR'20	**57.0**	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
SRDC [10]	CVPR'20	52.3	76.3	**81.0**	**69.5**	76.2	78.0	**68.7**	53.8	**81.7**	**76.3**	57.1	85.0	71.3
CDAN (Baseline) [6]	NeurIPS'18	55.6	72.5	77.9	62.1	71.2	73.4	61.2	52.6	80.6	73.1	55.5	81.4	68.1
CDAN+RADA	This work	56.5	**76.5**	79.5	68.8	**76.9**	**78.1**	66.7	**54.1**	81.0	75.1	**58.2**	**85.1**	**71.4**

Table 3: Performance (%) comparisons on Digit-Five with the state-of-the-art approaches for unsupervised domain adaptation (on the multi-source to single target adaptation settings). Cov_3FC_2 is taken as backbone for all these approaches. Note that we report the results of our baseline scheme CDAN [6] run by us where the original paper did not report results on this dataset.

Methods	Venue	Digit-Five					
		mt	*mm*	*sv*	*sy*	*up*	*Avg.*
DANN [4]	JMLR'16	97.9±0.83	70.8±0.94	68.5±0.85	87.3±0.68	93.4±0.79	83.6
IWAN [14]	ICML'19	98.2±0.13	74.2±0.22	72.9±0.56	88.9±0.54	95.8±0.51	86.0
MDAN [17]	NeurIPS'18	97.2±0.98	75.7±0.83	82.2±0.82	85.2±0.58	93.3±0.48	86.7
MCD [9]	CVPR'18	96.2±0.81	72.5±0.67	78.8±0.78	87.4±0.65	95.3±0.74	86.1
DCTN [12]	CVPR'18	99.4±0.06	76.2±0.51	86.8±0.31	94.4±0.58	86.4±0.54	88.6
M3SDA [7]	ICCV'19	98.4±0.68	72.8±1.13	81.3±0.86	89.5±0.56	96.1±0.81	87.6
CMSS [13]	ECCV'20	99.0±0.08	75.3±0.57	88.4±0.54	93.7±0.21	97.7±0.13	90.8
CDAN (Baseline) [6]	NeurIPS'18	99.1±0.28	71.3±0.23	85.2±0.21	90.1±0.37	97.8±0.22	88.7
CDAN+RADA	This work	**99.5±0.17**	**78.9±0.26**	**90.5±0.59**	**98.4±0.52**	**98.7±0.23**	**93.2**

Table 4: Performance (%) comparisons on DomainNet with the state-of-the-art approaches for unsupervised domain adaptation (on the multi-source to single target adaptation settings). All experiments are based on ResNet-101 pre-trained on ImageNet. Note that we report the results of our baseline scheme CDAN [6] run by us where the original paper did not report results on this dataset.

Methods	Venue	DomainNet						
		clipart	*infograph*	*painting*	*quickdraw*	*real*	*sketch*	*Avg*
DANN [4]	JMLR'16	45.5±0.59	13.1±0.72	37.0±0.69	13.2±0.77	48.9±0.65	31.8±0.62	32.65
DCTN [12]	CVPR'18	48.6±0.73	23.5±0.59	48.8±0.63	7.2±0.46	53.5±0.56	47.3±0.47	38.27
MCD [9]	CVPR'18	54.3±0.64	22.1±0.70	45.7±0.63	7.6±0.49	58.4±0.65	43.5±0.57	38.51
MDAN [17]	NeurIPS'18	60.3±0.41	25.0±0.43	50.3±0.36	8.2±1.92	61.5±0.46	51.3±0.58	42.80
M3SDA [7]	ICCV'19	58.6±0.53	26.0±0.89	52.3±0.55	6.3±0.58	62.7±0.51	49.5±0.76	42.67
FAR [5]	Arxiv'19	62.6±0.11	26.5±0.22	53.9±0.19	13.7±0.43	63.2±0.33	52.9±0.16	45.47
CMSS [13]	ECCV'20	64.2±0.18	**28.0±0.20**	53.6±0.39	16.0±0.12	63.4±0.21	53.8±0.35	46.52
CDAN (Baseline) [6]	NeurIPS'18	63.3±0.21	23.2±0.11	54.0±0.34	16.8±0.41	62.8±0.14	50.9±0.43	45.16
CDAN+RADA	This work	**66.9±0.33**	26.1±0.48	**54.6±0.19**	**18.9±0.35**	**63.9±0.36**	**54.6±0.18**	**47.50**