Supplementary Materials: Implicit Class-Conditioned Domain Alignment for Unsupervised Domain Adaptation

Xiang Jiang [1 2] **Qicheng Lao** [1 3] **Stan Matwin** [2 4] **Mohammad Havaei** [1]

A. Theory

Definition 1. *Let $\mathcal{B}_S$, $\mathcal{B}_T$ be minibatches from $\mathcal{U}_S$ and $\mathcal{U}_T$, respectively, where $\mathcal{B}_S \subseteq \mathcal{U}_S$, $\mathcal{B}_T \subseteq \mathcal{U}_T$, and $m_b = |\mathcal{B}_S| = |\mathcal{B}_T|$. The empirical estimation of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S, \mathcal{B}_T)$ over the minibatches $\mathcal{B}_S$, $\mathcal{B}_T$ is defined as*

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S, \mathcal{B}_T) = \frac{1}{m_b} \sup_{h,h' \in \mathcal{H}} \left| \sum_{\mathcal{B}_T} [h \neq h'] - \sum_{\mathcal{B}_S} [h \neq h'] \right|. \tag{1}$$

For simplicity, we drop the multiple $\frac{1}{m_b}$ in the following analysis as it does not affect the result of optimization.

Theorem 2 (The decomposition of $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S, \mathcal{B}_T)$)**.** *Let $\mathcal{H}$ be a hypothesis space and $\mathcal{Y}$ be the label space of the classification task where $\mathcal{B}_S$, $\mathcal{B}_T$ are minibatches drawn from $\mathcal{U}_S$, $\mathcal{U}_T$, respectively, and Y_S, Y_T are the label set of $\mathcal{B}_S$, $\mathcal{B}_T$. We define three disjoint sets on the label space: the shared labels $Y_C := Y_S \cap Y_T$, and the domain-specific labels $\overline{Y_S} := Y_S - Y_C$, and $\overline{Y_T} := Y_T - Y_C$. We also define the following disjoint sets on the input space where $\mathcal{B}_S^C := \{x \in \mathcal{B}_S \mid y \in Y_C\}$, $\mathcal{B}_S^{\overline{C}} := \{x \in \mathcal{B}_S \mid y \notin Y_C\}$, $\mathcal{B}_T^C := \{x \in \mathcal{B}_T \mid y \in Y_C\}$, $\mathcal{B}_T^{\overline{C}} := \{x \in \mathcal{B}_T \mid y \notin Y_C\}$. The empirical $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S, \mathcal{B}_T)$ divergence can be decomposed into class aligned divergence and class-misaligned divergence:*

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S, \mathcal{B}_T) = \sup_{h,h' \in \mathcal{H}} \left| \xi^C(h,h') + \xi^{\overline{C}}(h,h') \right|, \tag{2}$$

where

$$\xi^C(h,h') = \sum_{\mathcal{B}_T^C} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^C} \mathbb{1}[h \neq h'], \tag{3}$$

$$\xi^{\overline{C}}(h,h') = \sum_{\mathcal{B}_T^{\overline{C}}} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^{\overline{C}}} \mathbb{1}[h \neq h']. \tag{4}$$

Proof. By definition, we have

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S, \mathcal{B}_T) = \sup_{h,h' \in \mathcal{H}} \left| \sum_{\mathcal{B}_T} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S} \mathbb{1}[h \neq h'] \right| \tag{5}$$

We rewrite the summation over all the samples $\mathcal{B}$ into the sum of disjoint subsets $\mathcal{B}^C$ and $\mathcal{B}^{\overline{C}}$.

$$\sum_{\mathcal{B}_T} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S} \mathbb{1}[h \neq h'] \tag{6}$$

$$= \left(\sum_{\mathcal{B}_T^C} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^C} \mathbb{1}[h \neq h'] \right) \tag{7}$$

$$+ \left(\sum_{\mathcal{B}_T^{\overline{C}}} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^{\overline{C}}} \mathbb{1}[h \neq h'] \right) \tag{8}$$

$$= \xi^C(h,h') + \xi^{\overline{C}}(h,h'). \tag{9}$$

This completes the proof. □

B. Experiments

B.1. Additional Evaluation Measures on Office-Home

Table 1. Evaluation on Office-Home (%) with ResNet-50.

	Ar→Cl		Pr→Rw	
	MDD	ours	MDD	ours
accuracy	54.91	56.17	77.46	79.94
macro F1 score	53.66	55.29	75.86	78.42
weighted F1 score	53.97	55.81	77.24	79.79
macro precision	57.02	57.72	78.21	79.56
weighted precision	58.85	60.30	79.60	80.97
macro recall	56.41	57.76	76.30	78.61
weighted recall	54.91	56.17	77.65	79.94

Table 1 presents additional evaluation on Office-Home (standard). We re-implement MDD using identical batch sizes (50) and random seeds for fair comparison. The results show that our proposed method has consistent improvements across all evaluation measures, and the improvements are not a result of batch sizes or random seeds.

B.2. Additional Ablation on Alignment Options

Table 2 presents the ablation study on Office-Home (standard) that aims to assess the impact of different implicit alignment options: alignment in the domain divergence

Table 2. The impact of different implicit alignment options, i.e., masking the classifier-based domain discrepancy measure and sampling examples from the source and target domains, on Ar→Cl and Cl→Pr, Office-Home (standard).

	Alignment options		
Domains	masking	sampling	Accuracy
Ar→Cl	×	×	55.3
	√	×	55.5
	×	√	54.6
	√	√	**56.2**
Cl→Pr	×	×	71.4
	√	×	70.1
	×	√	70.5
	√	√	**73.1**

estimations (i.e., *masking* in MDD) and alignment in the input space (i.e., *sampling* class-conditioned examples). We observe that both alignment techniques are essential for domain adaptation because alignment should be enforced consistently across all aspects of the domain adaptation training. This is consistent to findings in the main paper.

B.3. Learning Curve

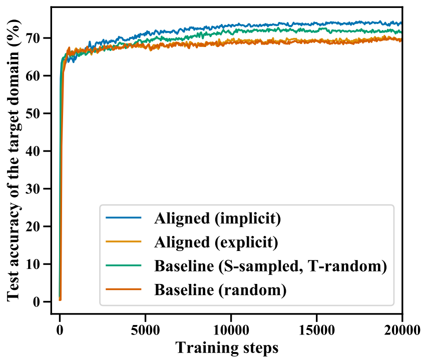

Figure 1. Learning curve of the target domain accuracy for Pr→Rw, Office-Home (RS-UT).

Figure 1 shows the learning curve of the target domain accuracy for different methods. The proposed implicit alignment converges better than other methods.

B.4. Computational Efficiency

Self-training requires estimating the target domain labels, which could be time-consuming depending on the size of the dataset. To improve the computational efficiency of our algorithm, we only update pseudo-labels periodically, i.e., every 20 steps, instead of at every training step. We show in Table 3 that different pseudo-label update frequencies exhibit similar performance on the target domain. Notably, implicit alignment outperforms the baseline method in spite of only updating the pseudo-labels every 500 training steps. This validates the robustness of implicit alignment.

Table 3. The impact of pseudo-label update frequency on Ar→Cl, Office-Home (standard).

pseudo-labels updated every N steps	accuracy
5	56.0
10	56.7
20	56.2
50	55.2
100	56.3
500	55.7

For the experiments described in Section B.3, training the baseline methods take 31 hours (wall clock time), whereas implicit alignment takes 34 hours under the same training condition when the pseudo-labels are updated every 20 steps. The 10% computational overhead is rather restricted. Moreover, from an engineering perspective, partially updating and caching the pseudo-labels could further improve the computational efficiency, and we leave them as future work.

B.5. Impact of Batch Size

Table 4. Impact of batch size on target domain accuracy (%), Ar→Cl, Office-Home (standard). The MDD results are based on our re-implementation.

batch size	baseline	implicit
8	48.9	49.7
16	52.7	52.8
32	54.9	56.2
50	55.3	56.2

Table 4 presents the impact of batch size on the target domain accuracy. We find that implicit alignment consistently improves the model performance over the MDD baseline across different batch sizes, and both methods work better with larger batch sizes.

B.6. Empirical Class Diversity

Figure 2 shows the empirical class diversity comparing implicit alignment with the baseline. In both experiments, the batch size is identical with the total number of classes (i.e., 31). For the baseline method, random sampling only obtains about 19 unique classes per-batch, which is much smaller than the batch size, in spite of the batch sizes being the

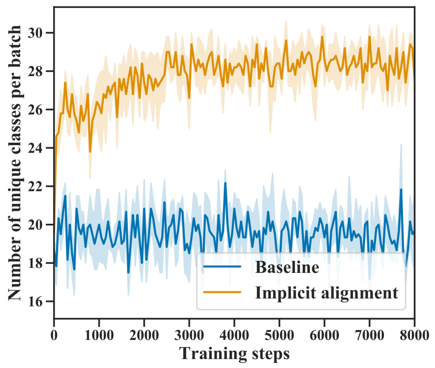

Figure 2. Empirical class diversity while training A→W (Office-31) with batch size 31.

same with the total number of classes. This is because random sampling can be viewed as sampling *with replacement* in the label space, whereas the implicit alignment can be viewed as sampling *without replacement* in the label space, which naturally increases the empirical class diversity. The expected class diversity of the baseline is

$$\mathbb{E}\left[|Y|\right] = n\left[1 - \left(\frac{n-1}{n}\right)^k\right], \tag{10}$$

where n is the number of unique classes and k is the size of the minibatch. The expected class diversity is 19.78 if $n = 31$ and $k = 31$, which is consistent with the empirical class diversity shown in Figure 2.

For the implicit alignment method shown in Figure 2, although it has low class diversity at training step 0 due to the random pseudo-labels, it has a sharp increase in class diversity for the first few hundred training steps, and eventually being able to sample 28 classes from the total of 31 classes. This confirms that implicit alignment is effective in improving empirical class diversity beyond random sampling.

C. Datasets

Figure 3 shows the frequencies of different classes for Cl→Rw on the Office-Home (standard) dataset. This dataset is under natrual class imbalance where examples of different classes are not evenly distributed.

Figure 4 shows the frequencies of different classes for Cl→Rw on the Office-Home (RS-UT) dataset (Tan et al., 2019). In this dataset, the minority classes in the source domain are majority classes in the target domain, which creates extreme within-domain class imbalance and between-domain distribution shift.

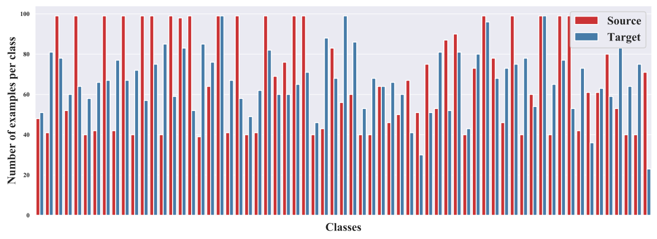

Figure 3. Class frequency of Cl→Rw, Office-Home (standard)

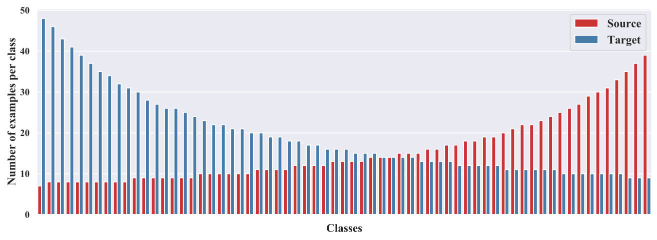

Figure 4. Class frequency of of Cl→Rw, Office-Home (RS-UT)

D. Model Architecture and Training Details

Code. We use PyTorch 1.2 as the training environment, and we observe that the adaptation performance on PyTorch 1.4 is slightly better PyTorch 1.2. The differences between PyTorch versions do not change the findings and the conclusions of this paper. Our code and training instructions are provided in `https://github.com/xiangdal/implicit_alignment`.

Model architecture. We use ResNet-50 (He et al., 2016) pre-trained from ImageNet (Russakovsky et al., 2015) as the backbone, and use hyper-parameters from (Zhang et al., 2019) for MDD-based domain discrepancy measure. The backbone is followed by a 1-1ayer bottleneck. The classifier f and auxiliary classifier f' are both 2-layer networks.

Optimization. We use the SGD optimizer with learning rate 0.001, nesterov momentum 0.9, and weight decay 0.0005. We empirically find that SGD converges better than Adam for adversarial optimization. We use a gradient reversal layer for minimax optimization, and we use a training scheduler (Ganin et al., 2016) for gradient reversal layer defined as

$$\lambda_p = \frac{0.2}{1 + \exp(-\frac{i}{1000})} - 0.1, \tag{11}$$

where i denotes the step number. We used the same scheduler from (Zhang et al., 2019) for all experiments and have not tried hyperparameter search for λ_p. The batch size is 31 for Office-31 and 50 for Office-Home.

References

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Tan, S., Peng, X., and Saenko, K. Generalized domain adaptation with covariate and label shift co-alignment. *arXiv preprint arXiv:1910.10320*, 2019.

Zhang, Y., Liu, T., Long, M., and Jordan, M. Bridging theory and algorithm for domain adaptation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7404–7413, Long Beach, California, USA, 09–15 Jun 2019. PMLR.