# Selective Partial Domain Adaptation

Pengxin Guo
12032913@mail.sustech.edu.cn
Jinjing Zhu
zhujinjing.hust@gmail.com
Yu Zhang[†]
yu.zhang.ust@gmail.com

Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China

**Abstract**

Partial Domain Adaptation (PDA), which assumes that the label space of the target domain is a subset of that in the source domain, has attracted much attention in recent years. Due to the difference in the label space of these two domains, it is hard to directly align these two domains in PDA. To solve this problem, we propose a Selective Partial Domain Adaptation (SPDA) method, which selects useful data for the adaptation to the target domain. Specifically, we firstly design a Maximum of Cosine (MoC) similarity function customized for PDA to select useful data in the source domain to decrease the domain discrepancy. In the MoC similarity function, for each target sample, we select the source sample with the maximal cosine similarity for adaptation. Moreover, a selective training method is designed to add useful target data into the source domain. In detail, the selective training method firstly assigns pseudo-labels to target samples with the self-training strategy and then adds target samples with high confidence in terms of pseudo-labels to the source domain. Based on these two selection operations, the proposed SPDA method can select useful data for domain adaptation. Experiments on several datasets demonstrate the effectiveness of the proposed SPDA method.

## 1 Introduction

Deep neural networks have achieved great performance on a variety of computer vision problems [6, 11, 12, 19, 24, 35]. However, the great performance benefits from a large amount of labeled data, which is not easy to obtain in many real-world applications. Since it is laborious to manually label sufficient training data for various applications, Domain Adaptation (DA) [20, 34] is proposed to solve this problem. DA aims to transfer knowledge learned in a data-abundant source domain to help the learning in a target domain with only a large amount of unlabeled data.

Most existing DA methods assume that the target domain shares an identical label space to the source domain but with different data distributions. However, in real-world applications, it is usually not easy to find such a source domain for a target domain. Hence, Partial Domain Adaptation (PDA) studies a relaxed setting, where the label space of the target domain is a subset of that of the source domain.

[†] Corresponding author

The core problem in both DA and PDA is that there exits a distribution shift across domains, which hinders the direct generalization of the source model to the target domain. In DA, one solution to this issue is to learn a domain-invariant feature representation which could be discriminative for classification. To learn such a feature representation, many distance functions have been adopted in existing DA methods, such as the Maximum Mean Discrepancies (MMD) [9] used in the Deep Domain Confusion (DDC) [30] and Deep Adaptation Network (DAN) [17], the Kullback-Leibler (KL) divergence adopted in the Transfer Learning with Deep Autoconders (TLDA) [39], the second-order statistics utilized in the CORrelation ALignment (CORAL) [28, 29], and the Central Moment Discrepancy (CMD) proposed in [36]. However, due to the difference in the label spaces of the two domains, all of these distance functions are inapplicable to PDA. If we forcibly utilize these distance functions to align the source and target domains in PDA, it may lead to negative transfer [2].

To learn under the PDA setting, a possible way is to select useful source samples whose labels are highly likely to appear in the target domain for the adaptation. However, since the target domain is unlabelled, it is not straightforward to identify which classes are presented in the target domain and which source samples are helpful for the target domain. To solve those issues, in this paper, we propose a Selective Partial Domain Adaptation (SPDA) method. Specifically, we firstly design a Maximum of Cosine (MoC) similarity function customized for PDA to select the source sample with the maximal cosine similarity for each target sample. In this way, we can select the most useful source samples for adaptation and ignore irrelevant source samples which may cause negative transfer. An illustration of the MoC similarity is shown in Fig. 1. Furthermore, to fully exploit target samples, we adopt a selective training method in the proposed SPDA model. In detail, the self-training strategy is first used to assign pseudo-labels to target samples. Then the target samples with high confidence are selected to add to the source domain. In this way, the proposed SPDA method not only makes the model generalize better to the target domain due to the use of the target data with high confidence pseudo-labels but also maximizes the intra-domain similarity in the target domain since each target sample can select a target sample, which has been added into the source domain, to maximize their similarity based on the proposed MoC similarity. Experiments on several benchmark datasets demonstrate the effectiveness of the proposed SPDA method. In summary, our contributions are three-fold as summarized in the following.

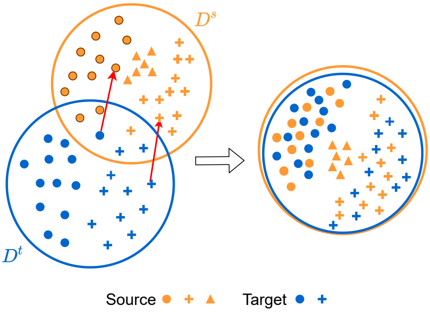

Figure 1: Illustration for the MoC similarity. Since in PDA the label space of the target domain is a subset of that of the source domain, we cannot align these two domains directly. In the MoC similarity function, for each target sample, we select the source sample with the maximal cosine similarity and hope to draw them closely. After that, the source and target domains could be well aligned. *Best viewed in color.*

- We propose the MoC similarity, which can select useful source samples for adaptation under the PDA setting even while the target domain is unlabeled.
- Built on the MoC similarity, we propose the selective training method as another selec-

tion operation in SPDA to choose target samples with high confidence pseudo-labels and add these data to the source domain to maximize the intra-domain similarity in the target domain.

- Extensive experiments are conducted on three PDA benchmark datasets, including Office-31, Office-Home, and VisDA-2017, to show the superiority of the proposed SPDA method over state-of-the-art DA and PDA methods.

## 2 Related Work

**Partial Domain Adaptation** In the PDA scenario, the label space of the target domain is a subset of that in the source domain. Cao *et al.* [2] firstly introduce the PDA setting and propose the Partial Adversarial Domain Adaptation (PADA) method to simultaneously circumvent negative transfer and promote positive transfer. Then several models are proposed to solve the PDA problem, including the Importance Weighted Adversarial Network (IWAN) [37] which utilizes a weighting scheme based on adversarial networks, Example Transfer Network (ETN) [3] which quantifies the transferability of source samples and discovers shared label space, Deep Residual Correction Network (DRCN) [15] which adds a residual block into the source network along with the task-specific feature layer to effectively enhance the adaptation, Reinforced Transfer Network (RTNet) [4] which adopts reinforcement learning to automatically select source samples in the shared classes, BA$^3$US [16] which designs the balanced adversarial alignment and adaptive uncertainty suppression to conduct uncertainty propagation, Selective Representation Learning for Class-Weight Computation (SRLCWC) [5] which first identifies outlier classes based on the image content information and then trains a label classifier on the class content from source images, confused classes, and Domain Consensus Clustering (DCC) [14] which exploits the domain consensus knowledge to discriminate common classes from private classes and then determines clusters as well as private classes, Implicit Semantic Response Alignment [33] which boosts the existing partial domain adaptation models by exploring inherent class relationship across both source and target domains, and Adversarial Reweighting (AR) [10] which adversarially learns the weights of source domain data to align the source and target domain distributions. Different from the aforementioned methods, in this paper, we propose the SPDA method to select useful source samples based on the proposed MoC similarity function.

**Self-training** The self-training strategy aims at iteratively training the model by using both labeled data and unlabeled data with assigned pseudo-labels, and it is initially explored in semi-supervised learning [8, 38]. Recently, some works [13, 18, 23, 27, 40, 41] apply the self-training strategy to DA. For example, Zou *et al.* [40] formulate the "domain gap" problem in DA as a latent variable and solve it via an iterative self-training strategy. Zou *et al.* [41] propose a confidence regularized self-training (CRST) framework to treat pseudo-labels as continuous latent variables that are jointly optimized with model parameters. Mei *et al.* [18] develop a pseudo-label generation strategy with an instance-adaptive selector to effectively improve the quality of pseudo-labels and propose an instance adaptive self-training framework for DA on the semantic segmentation task. Kumar *et al.* [13] use the self-training strategy to solve the gradual domain adaptation problem. Phoo and Hariharan [23] self-train on unlabeled target samples to tackle the extreme domain gap in DA. However, all of the above methods are proposed for DA, where the target and source domains share an identical label space. In this paper, we consider the PDA setting, where the label space of the target domain belongs to that of the source domain. Moreover, the use of target samples and pseudo-labels in the proposed method is different from the aforementioned existing works. Specifically, we not only use the pseudo-labels of target samples to compute the classification loss, but also use them to compute the MoC loss by the proposed selective training method, which can reduce both inter-domain and intra-domain gaps.

# 3 SPDA

In this section, we introduce the proposed SPDA method under the PDA setting.

In the PDA setting, we have a labeled source dataset $\mathcal{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and an unlabeled target dataset $\mathcal{D}_t = \{x_t^i\}_{i=1}^{n_t}$, where $n_s$ and $n_t$ denote the number of samples in the source and target domains, respectively. These two domains have different data distributions, i.e., $p_s(x_s) \neq p_t(x_t)$, due to the domain shift. Notably, in PDA, the target label space is a subset of the source label space, i.e., $\mathcal{Y}_t \subset \mathcal{Y}_s$. The goal is to train a model that can utilize useful knowledge in the source domain $\mathcal{D}_s$ to help the learning in the target domain $\mathcal{D}_t$.

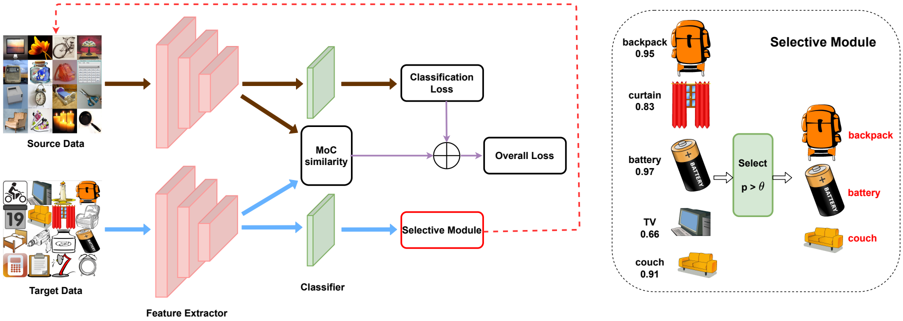

Figure 2: Illustration of the proposed SPDA method. The left figure shows the whole architecture of the SPDA model, whose objective function consists of two parts, including the classification loss on the source data as well as the selected target data with high confidence pseudo-labels and the negative MoC similarity between augmented source samples and target samples. Note that the source and target networks share the same architecture and parameters. The right figure shows the selective module used for the target data, where only target samples with high confidence pseudo-labels will be added to the source domain.

As shown in Fig. 2, the proposed SPDA method consists of two selection operations. The first selection operation is to design the MoC similarity to select useful source samples for adaptation and the second one is to utilize the selective training method to select target samples with high confidence pseudo-labels and add them to the source domain. In the following, we first introduce these two parts and then present the overall objective function of the SPDA method.

## 3.1 MoC Similarity

The main problem of PDA is the difference between label spaces of the two domains. However, existing distance functions adopted in DA such as MMD, KL divergence, CORAL, and

CMD do not take this problem into consideration, causing to suffer from the negative transfer with high probability due to irrelevant source samples. To provide a remedy to this problem, we design the MoC similarity function customized for PDA to measure the similarity between the source and target domains. Specifically, for each sample in the target domain, we select the source sample with the maximal cosine similarity to define the similarity between the two domains. The MoC similarity function $\mathrm{MoC}(X_S, X_T)$ is formulated as

$$\mathrm{MoC}(X_S, X_T) = \frac{1}{n_t} \sum_{j=1}^{n_t} \max_{i \in [n_s]} \frac{(G(x_s^i))^\top G(x_t^j)}{\|G(x_s^i)\|_2 \|G(x_t^j)\|_2}, \tag{1}$$

where $X_S$ and $X_T$ denote the source and target datasets, respectively, $[n] = \{1, 2, \cdots, n\}$ denotes the set of positive integers up to an integer $n$, $\|\cdot\|_2$ denotes the $L_2$ norm, and $G(\cdot)$ denotes the feature extraction network used in SPDA. Note that the MoC similarity is computed based on the hidden feature representation but not the original data representation. The MoC similarity measures the domain similarity based on each target sample and the most similar source sample in terms of the cosine similarity. When maximizing the MoC similarity, such a pair of samples from the two domains will become more similar and the two domains could be well aligned. Note that in the MoC strategy, we only select samples with maximum cosine similarities, which implies that these two samples are very likely to be related (i.e., they either belong to the same class or have similar semantic representations), which can guarantee that the selected samples are useful for aligning two domains. Moreover, in the implementation, the MoC criterion is calculated based on each mini-batch including source and target samples and its complexity is proportional to the product between numbers of source and target samples in a mini-batch, which is training efficient.

The value range of the MoC similarity is in $[-1, 1]$. The larger the MoC similarity, the more similar the two domains. When the MoC similarity equals the maximum 1, for each target sample, we can find at least one source sample that is almost identical to the target sample, which means that the distribution of the target domain is close to some part of the source distribution. When the MoC similarity equals the minimum $-1$, the target domain is dissimilar to the source domain. In this case, the best choice is not to use the source data for adaptation.

## 3.2 Selective Training

In the following, we introduce the proposed selective training method adopted in the proposed SPDA method.

Specifically, we first generate pseudo-labels for target samples by the self-training strategy in each training epoch. The process of generating pseudo-labels can be formulated as

$$\{\hat{y}_t, p_t\} = \max(F(G(x_t))), \tag{2}$$

where $F(\cdot)$ denotes the classification network, and the max operation will return the index and value of the maximal value, which means that $\hat{y}_t$ represents the pseudo-label and $p_t$ represents the confidence. Then we select those target samples with high confidence pseudo-labels as

$$\hat{X}_T = \{x_t^j \mid p_t^j > \theta, \forall j \in [n_t]\}, \tag{3}$$

where $\theta$ denotes a threshold to determine whether a pseudo-label is of high confidence. Then those selected target samples are considered to be similar to the source domain and added to the source domain. We denote the augmented source domain by

$$\widetilde{X}_S = \{X_S, \hat{X}_T\}. \tag{4}$$

Such augmented source domain can not only provide more labeled data to learn a more accurate learner for the target domain but also increase the intra-domain similarity in the target domain as shown in the next section. Note that the pseudo-labels of target samples, selected target data $\hat{X}_T$, and augmented source data $\hat{X}_S$ are updated at each training epoch.

## 3.3 Overall Objective Function

By combining these two selection operations, the overall objective function of the proposed SPDA method is formulated as

$$\min_{\mathbf{w}} \mathcal{L}_C(\widetilde{X}_S, \{y_s, \hat{y}_t\}) - \lambda \mathrm{MoC}(\widetilde{X}_S, X_T), \tag{5}$$

where $\mathbf{w}$ denotes parameters of the whole network that consists of $G$ and $F$, $\mathcal{L}_C(\widetilde{X}_S, \{y_s, \hat{y}_t\})$ denotes the classification loss on the labeled source samples $X_S$ with their ground truth labels $y_s$ and the selected target samples $\hat{X}_T$ with their pseudo-labels $\hat{y}_t$, and $\lambda$ is a hyperparameter to balance the two terms in problem (5). Since a large MoC similarity indicates a similar pair of the source and target domains, we aim to maximize $\mathrm{MoC}(\widetilde{X}_S, X_T)$ or equivalently minimize the negative of $\mathrm{MoC}(\widetilde{X}_S, X_T)$ in problem (5). The first term in problem (5) (i.e., $\mathcal{L}_C(\widetilde{X}_S, \{y_s, \hat{y}_t\})$) based on the cross-entropy loss consists of two parts as

$$\mathcal{L}_C(\widetilde{X}_S, \{y_s, \hat{y}_t\}) = \mathcal{L}_C(X_S, y_s) + \mathcal{L}_C(\hat{X}_T, \hat{y}_t), \tag{6}$$

where $\mathcal{L}_C(X_S, y_s)$, $\mathcal{L}_C(\hat{X}_T, \hat{y}_t)$ are defined as

$$\mathcal{L}_C(X_S, y_s) = -\frac{1}{n_s + \hat{n}_t} \sum_{i=1}^{n_s} (z_s^i)^T \log F(G(x_s^i)),$$

$$\mathcal{L}_C(\hat{X}_T, \hat{y}_t) = -\frac{1}{n_s + \hat{n}_t} \sum_{j=1}^{\hat{n}_t} (z_t^j)^T \log F(G(\hat{x}_t^j)),$$

where $\hat{n}_t$ denotes the number of samples in $\hat{X}_T$, $\hat{x}_t^j$ denotes the $j$th sample in $\hat{X}_T$, $z_s^i$ denotes the one-hot label vector corresponding to $x_s^i$, $z_t^j$ is the one-hot label vector for $\hat{x}_t^j$.

According to problem (5), the augmented source domain $\tilde{X}_S$ contributes to both terms. Firstly, the selected target samples in $\widetilde{X}_S$ can be helpful to learn the classifier as it brings additional supervision information as shown in Eq. (6). Secondly, it is easy to write $\mathrm{MoC}(\widetilde{X}_S, X_T)$ as

$$\mathrm{MoC}(\widetilde{X}_S, X_T) = \frac{1}{n_t} \sum_{j=1}^{n_t} \max\left( \max_{i \in [n_s]} \cos(x_t^j, x_s^i), \max_{k \in [\hat{n}_t]} \cos(x_t^j, \hat{x}_t^k) \right), \tag{7}$$

where $\cos(\cdot,\cdot)$ denotes the cosine similarity between two samples as defined in Eq. (1). Compared with $\mathrm{MoC}(X_S, X_T)$, $\mathrm{MoC}(\widetilde{X}_S, X_T)$ may use the cosine similarity between a target sample and a target sample in $\hat{X}_T$ to define the domain similarity. Thus maximizing $\mathrm{MoC}(\widetilde{X}_S, X_T)$ may maximize the intra-domain similarity for the target domain.

Based on the above analysis, both the MoC similarity and the selective training method are important blocks for the SPDA method, which will be verified in Section 4.3. The MoC similarity can measure the domain similarity and help align two domains, while the selective training method provides more labeled data from the target domain.

# 4 Experiments

In this section, we conduct experiments on three benchmark datasets (i.e., Office-31 [26], Office-Home [32], and VisDA-2017 [22]) to evaluate the proposed SPDA method. Due to page limit, the detailed introduction of benchmark datasets, baseline methods, experimental setup, and some experimental results are put in the appendix. The code is available at https://github.com/gpx333/SPDA.

## 4.1 Why Use the Cosine Similarity?

Before presenting experimental results and analyses, we first explain why the cosine similarity in the MoC similarity is suitable.

To see the effect of the cosine similarity used in the MoC similarity, we compare with different popular distance functions, including the Euclidean Distance (ED), Manhattan Distance (MD), and Chebyshev Distance (CD). Specifically, we replace the cosine similarity by those distance functions in Eq. (1) and by taking the Euclidean distance as an example, the corresponding distance function is formulated as

$$\mathrm{ED}(X_S, X_T) = \frac{1}{n_t} \sum_{j=1}^{n_t} \min_{i \in [n_s]} \|G(x_s^i) - G(x_t^j)\|_2. \tag{8}$$

As we need to minimize those distance functions to make two domains aligned, the objective function for the Euclidean distance is formulated as

$$\min_{\mathbf{w}} \mathcal{L}_C(\widetilde{X}_S, \{y_s, \hat{y}_t\}) + \lambda \mathrm{ED}(\widetilde{X}_S, X_T),$$

which differs from problem (5) in the second term.

Then we conduct experiments on three hard transfer tasks (i.e., Ar→Cl, Pr→Cl, and Rw→Cl), whose hardness is measured by the transfer performance, on the Office-Home dataset to compare the MoC similarity with those distance functions. According to the results shown in Table 1, we can see that all the distance functions (i.e., ED, MD, and CD) perform inferior to the MoC similarity. One possible reason is that in a high-dimensional space, distance functions cannot measure the difference of two hidden representations accurately. Hence, we adopt the cosine similarity in the proposed SPDA method.

| Method | Ar→Cl | Pr→Cl | Rw→Cl | Avg |
|---|---|---|---|---|
| SPDA-ED | 37.37 | 34.57 | 41.79 | 37.91 |
| SPDA-MD | 35.16 | 35.40 | 34.03 | 34.86 |
| SPDA-CD | 46.81 | 39.40 | 43.22 | 43.14 |
| SPDA-MoC | 64.24 | 58.91 | 67.41 | 63.52 |

Table 1: Accuracy (%) on three transfer tasks of Office-Home with different distance functions and the proposed MoC similarity.

## 4.2 Results

According to the results on the Office-31, Office-Home, and VisDA-2017 datasets as shown in Tables 2, 3, and 4, we can see that the proposed SPDA method performs better or at least comparable to all the baseline methods on average and on all the tasks, which demonstrates the effectiveness of the SPDA method.

| Method | A→W | D→W | W→D | A→D | D→A | W→A | Avg |
|---|---|---|---|---|---|---|---|
| ResNet-50 [11] | 75.59±1.09 | 96.27±0.85 | 98.09±0.74 | 83.44±1.12 | 83.92±0.95 | 84.97±0.86 | 87.05 |
| DAN [17] | 59.32±0.49 | 73.90±0.38 | 90.45±0.36 | 61.78±0.56 | 74.95±0.67 | 67.64±0.29 | 71.34 |
| DANN [7] | 73.56±0.15 | 96.27±0.26 | 98.73±0.20 | 81.53±0.23 | 82.78±0.18 | 86.12±0.15 | 86.50 |
| ADDA [31] | 75.67±0.17 | 95.38±0.23 | 99.85±0.12 | 83.41±0.17 | 83.62±0.14 | 84.25±0.13 | 87.03 |
| PADA [2] | 86.54±0.31 | 99.32±0.45 | **100.0**±0.00 | 82.17±0.37 | 92.69±0.29 | 95.41±0.33 | 92.69 |
| IWAN [37] | 89.15±0.37 | 99.32±0.32 | 99.36±0.24 | 90.45±0.36 | 95.62±0.29 | 94.26±0.25 | 94.69 |
| SAN [1] | 93.90±0.45 | 99.32±0.52 | 99.36±0.12 | 94.27±0.28 | 94.15±0.36 | 88.73±0.44 | 94.96 |
| ETN [3] | 94.52±0.20 | **100.0**±0.00 | **100.0**±0.00 | 95.03±0.22 | 96.21±0.27 | 94.64±0.24 | 96.73 |
| RTNet [4] | 96.20±0.30 | **100.0**±0.00 | **100.0**±0.00 | 97.60±0.10 | 92.30±0.10 | 95.40±0.10 | 96.90 |
| BA$^3$US [16] | 98.98±0.28 | **100.0**±0.00 | 98.73±0.00 | **99.36**±0.00 | 94.82±0.05 | 94.99±0.08 | 97.81 |
| DRCN [15] | 88.05 | **100.0** | **100.0** | 86.00 | 95.60 | 95.80 | 94.30 |
| SRLCWC [5] | 92.07 | 95.84 | 99.24 | 94.46 | 93.68 | 93.72 | 94.84 |
| DCC [14] | **99.70** | **100.0** | **100.0** | 96.10 | 95.30 | 96.30 | 97.90 |
| SPDA (Ours) | 99.32±0.02 | **100.0**±0.00 | **100.0**±0.00 | 96.18±0.32 | 96.03±0.25 | **96.56**±0.00 | **98.01** |

Table 2: Accuracy (%) on the Office-31 dataset under the PDA setting with the ResNet-50 as the backbone.

| Method | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet-50 [11] | 46.33 | 67.51 | 75.87 | 59.14 | 59.94 | 62.73 | 58.22 | 41.79 | 74.88 | 67.40 | 48.18 | 74.17 | 61.35 |
| DAN [17] | 35.70 | 52.90 | 63.70 | 45.00 | 51.70 | 49.30 | 42.40 | 31.50 | 68.70 | 59.70 | 34.60 | 67.80 | 50.30 |
| DANN [7] | 43.76 | 67.90 | 77.47 | 63.73 | 58.99 | 67.59 | 56.84 | 37.07 | 76.37 | 69.15 | 44.30 | 77.48 | 61.72 |
| ADDA [31] | 45.23 | 68.79 | 79.21 | 64.56 | 60.01 | 68.29 | 57.56 | 38.89 | 77.45 | 70.28 | 45.23 | 78.32 | 62.82 |
| PADA [2] | 51.95 | 67.00 | 78.74 | 52.16 | 53.78 | 59.03 | 52.61 | 43.22 | 78.79 | 73.73 | 56.6 | 77.09 | 62.06 |
| IWAN [37] | 53.94 | 54.45 | 78.12 | 61.31 | 47.95 | 63.32 | 54.17 | 52.02 | 81.28 | 76.46 | 56.75 | 82.90 | 63.56 |
| SAN [1] | 44.42 | 68.68 | 74.60 | 67.49 | 64.99 | 77.80 | 59.78 | 44.72 | 80.07 | 72.18 | 50.21 | 78.66 | 65.30 |
| ETN [3] | 59.24 | 77.03 | 79.54 | 62.92 | 65.73 | 75.01 | 68.29 | 55.37 | 84.37 | 75.72 | 57.66 | 84.54 | 70.45 |
| RTNet [4] | 63.20±0.10 | 80.10±0.20 | 80.70±0.10 | 66.70±0.10 | 69.30±0.20 | 77.20±0.20 | 71.60±0.30 | 53.90±0.30 | 84.60±0.10 | 77.40±0.20 | 57.90±0.30 | 85.50±0.10 | 72.30 |
| BA$^3$US [16] | 60.62±0.45 | 83.16±0.12 | 88.39±0.19 | 71.75±0.19 | 72.79±0.19 | **83.40**±0.59 | 75.45±0.19 | 61.59±0.37 | 86.53±0.22 | 79.25±0.65 | 62.80±0.51 | 86.05±0.26 | 75.98 |
| DRCN [15] | 54.00 | 76.40 | 83.00 | 62.10 | 64.50 | 71.00 | 70.80 | 49.80 | 80.50 | 77.50 | 59.10 | 79.90 | 69.00 |
| SRLCWC [5] | 56.21 | 73.34 | 80.63 | 64.08 | 61.72 | 66.41 | 70.83 | 53.13 | 83.57 | 77.01 | 58.31 | 81.24 | 68.87 |
| DCC [14] | 59.00 | 84.40 | 83.40 | 67.80 | 72.70 | 79.80 | 68.40 | 53.20 | 83.70 | 75.80 | 59.00 | **88.30** | 73.00 |
| SPDA (Ours) | **64.24**±0.24 | **87.79**±0.11 | **88.74**±0.08 | **74.29**±0.22 | **75.10**±0.03 | 79.05±0.33 | **79.37**±0.15 | 58.91±0.13 | 85.05±0.42 | **81.36**±0.09 | **67.41**±0.21 | 84.09±0.38 | **77.12** |

Table 3: Accuracy (%) on the Office-Home dataset under the PDA setting with the ResNet-50 as the backbone.

On the Office-31 dataset, as Table 2 shows, all the DA methods (i.e., DAN, DANN, and ADDA) are inferior to the standard ResNet-50, showing that they suffer from the negative transfer, which means DA methods cannot be directly applied to the PDA setting due to the difference in the label spaces of the two domains. Moreover, the proposed SPDA method achieves the best average accuracy and performs the best in three out of six transfer tasks. Specifically, for easy transfer tasks (i.e., D→W and W→D), similar to some baseline methods, the proposed SPDA method achieves 100% accuracy and is very stable with zero standard deviations. For hard transfer tasks (i.e., D→A and W→A), the proposed SPDA method achieves the best or the second best results, which shows that our method is suitable for hard transfer tasks that transfer from a small domain to a large domain.

On the large-scale challenging Office-Home dataset, the proposed SPDA method outperforms all the baseline methods and obtains the best average accuracy 77.12%, which is about 1.2% higher than the state-of-the-art performance achieved by the BA$^3$US method as shown in Table 3. In all the 12 transfer tasks on this dataset, the proposed SPDA method achieves the best results in eight tasks. Similar to the Office-31 dataset, for transfer tasks from a small domain to a large domain (i.e., Ar→Cl, Ar→Pr, and Ar→Rw), the proposed SPDA method achieves the best results, showing that the SPDA method is suitable for this setting. For all the DA methods, DAN performs inferior to ResNet-50, leading to the negative transfer, while DANN and ADDA perform comparable to ResNet-50, implying that DA methods may not be helpful on this dataset.

On the most challenging VisDA-2017 dataset, as shown in Table 4, the proposed SPDA method achieves new state-of-the-art results on the two tasks and on average, which improves the current best results (i.e., 78.24% (R→S) by ETN, 74.27% (S→R) by BA$^3$US, and 73.39% (Avg) by ETN) by a large margin of about **14.23**%, **8.64**%, and **14.30**%, respectively. One possible reason of such significant improvement made by the proposed method is that the VisDA-2017 dataset is so large that existing methods may be hard to choose appropriate samples, while the proposed method can utilize the proposed MoC to do a better job.

| Method | R→S | S→R | Avg |
|---|---|---|---|
| ResNet-50 [11] | 64.28 | 45.26 | 54.77 |
| DAN [17] | 68.35 | 47.60 | 57.98 |
| DANN [7] | 73.84 | 51.01 | 62.43 |
| PADA [2] | 76.50 | 53.53 | 65.01 |
| IWAN [37] | 71.30 | 48.60 | 59.95 |
| SAN [1] | 69.70 | 49.90 | 59.80 |
| ETN [3] | *78.24* | *68.53* | *73.39* |
| BA$^3$US [16] | *69.25* | *74.27* | *71.76* |
| DRCN [15] | 73.20 | 58.20 | 65.70 |
| SPDA (Ours) | **92.47**±3.83 | **82.91**±1.76 | **87.69** |

Table 4: Accuracy (%) on the VisDA-2017 dataset under the PDA setting with the ResNet-50 as the backbone.

## 4.3 Ablation Study

We conduct ablation study on the Office-Home and Office-31 datasets to analyze the effects of the MoC similarity and the selective training method used in the SPDA method, respectively.

| Method | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 46.33 | 67.51 | 75.87 | 59.14 | 59.94 | 62.73 | 58.22 | 41.79 | 74.88 | 67.40 | 48.18 | 74.17 | 61.35 |
| SPDA w/o ST | 56.06 | 77.48 | 82.61 | 67.68 | 64.93 | 73.50 | 70.71 | 50.51 | 81.56 | 79.98 | 61.49 | 82.86 | 70.78 |
| SPDA w/o MoC | 60.66 | 85.27 | 86.25 | 69.97 | 70.14 | 77.03 | 70.71 | 52.00 | 81.23 | 76.68 | 56.36 | 79.22 | 72.13 |
| SPDA | 64.24 | 87.79 | 88.74 | 74.29 | 75.10 | 79.05 | 79.37 | 58.91 | 85.05 | 81.36 | 67.41 | 84.09 | 77.12 |

Table 5: Ablation study on the Office-Home dataset, where 'ST' denotes the selective training method.

From the ablation study on the Office-Home dataset shown in Table 5, in some transfer tasks (i.e., Pr→Rw, Rw→Ar, Rw→Cl, Rw→Pr), the performance of SPDA without the MoC similarity is inferior to SPDA without the selective training method, which means the MoC similarity is more important than the selective training method in these transfer tasks. In other transfer tasks, the selective training method seems to be more important than the MoC similarity. Furthermore, using only the selective training method (i.e., SPDA w/o MoC) and using only the MoC similarity (i.e., SPDA w/o ST) perform better than ResNet-50, which shows the effectiveness of these two parts. When these two parts are used together (i.e., SPDA), the best performance is achieved, which proves the importance of these two parts to the proposed SPDA method.

Results of the ablation study on the Office-31 dataset are shown in Fig. 3. According to the results, similar observations to the Office-Home dataset are observed, which again demonstrates the importance of these two parts to the SPDA method.

## 4.4 Sensitivity Analysis

We conduct sensitivity analysis with respect to the threshold $\theta$ on transfer tasks: A→W on the Office-31 dataset and Ar→Cl on the Office-Home dataset. In detail, we vary the

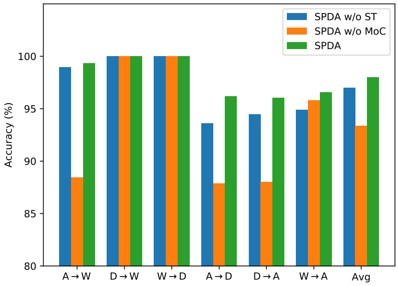

Figure 3: Ablation study on the Office-31 dataset, where ‘ST’ denotes the selective training method.

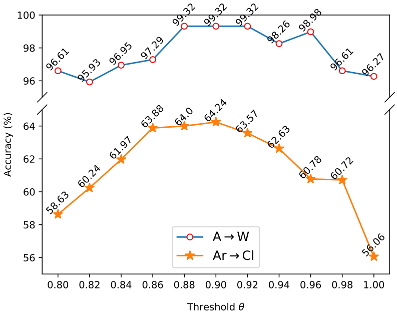

Figure 4: The performance of the SPDA method on two transfer tasks A→W and Ar→Cl when varying with the threshold $\theta$.

threshold $\theta$ from 0.8 to 1.0 at an interval of 0.02 with the experimental results shown in Fig. 4. According to the results, we can see that when $\theta$ is between 0.88 and 0.92, the performance does not make much difference, which implies that the SPDA method is not so sensitive to $\theta$ within a certain range. Moreover, when $\theta$ equals 0.9, the SPDA method achieves the best performance on these two transfer tasks. Hence, we set $\theta$ to 0.9 in all the experiments.

## 5 Conclusions

In this paper, we propose the SPDA method, which selects useful data for partial domain adaptation. Specifically, we first design the MoC similarity function for PDA to select useful source data to measure the domain similarity. Then, we propose the selective training method to first select target data with high confidence pseudo-labels and then add these data to the source domain. Experiments on several datasets demonstrate the effectiveness of our method. In future studies, we are interested in applying the MoC similarity to other DA settings.

## Acknowledgements

This work is supported by NSFC general grant 62076118 and Shenzhen fundamental research program JCYJ20210324105000003.

## References

[1] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2724–2732, 2018.

[2] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial do-

main adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[3] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2985–2994, 2019.

[4] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. Selective transfer with reinforced transfer network for partial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12706–12714, 2020.

[5] Sandipan Choudhuri, Riti Paul, Arunabha Sen, Baoxin Li, and Hemanth Venkateswara. Partial domain adaptation using selective representation learning for class-weight computation. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pages 289–293. IEEE, 2020.

[6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[7] 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.

[8] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

[9] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520, 2006.

[10] Xiang Gu, Xi Yu, Jian Sun, Zongben Xu, et al. Adversarial reweighting for partial domain adaptation. *Advances in Neural Information Processing Systems*, 34:14860–14872, 2021.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[13] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020.

[14] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9757–9766, 2021.

[15] Shuang Li, Chi Harold Liu, Qiuxia Lin, Qi Wen, Limin Su, Gao Huang, and Zhengming Ding. Deep residual correction network for partial domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[16] Jian Liang, Yunbo Wang, Dapeng Hu, Ran He, and Jiashi Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 123–140. Springer, 2020.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[18] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 415–430. Springer, 2020.

[19] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[20] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[22] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[23] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences. *arXiv preprint arXiv:2010.07734*, 2020.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115 (3):211–252, 2015.

[26] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[27] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *European conference on computer vision*, pages 532–548. Springer, 2020.

[28] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[29] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[30] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[31] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[32] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[33] Wenxiao Xiao, Zhengming Ding, and Hongfu Liu. Implicit semantic response alignment for partial domain adaptation. *Advances in Neural Information Processing Systems*, 34:13820–13833, 2021.

[34] Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer learning*. Cambridge University Press, 2020.

[35] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

[36] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

[37] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8156–8164, 2018.

[38] Xiaojin Zhu. Semi-supervised learning tutorial. In *International conference on machine learning (ICML)*, pages 1–135, 2007.

[39] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[40] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.

[41] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.

# A Experiments

## A.1 Setup

The Office-31 dataset is one of the most widely used datasets for visual domain adaptation. It has 4,652 images from 31 classes collected in three distinct domains: *Amazon* (**A**), *Webcam* (**W**), and *DSLR* (**D**). By following the protocol in [2], we select images from the 10 classes shared by Office-31 and Caltech-256 to build a target domain, and create six PDA tasks: **A→W**, **D→W**, **W→D**, **A→D**, **D→A**, and **W→A**. Note that there are 31 classes in the source domain and 10 classes in the target domain.

The Office-Home dataset is a better organized but more difficult dataset than the Office-31 dataset and it consists of 15,500 images in 65 object classes under the office and home settings, leading to four extremely dissimilar domains: *Artistic* (**Ar**), *Clip Art* (**Cl**), *Product* (**Pr**), and *Real-World* (**Rw**). For the PDA setting, we follow [2] to select images from the first 25 classes in an alphabetical order as the target domain and images from all 65 classes as the source domain, and hence obtain 12 PDA tasks: **Ar→Cl**, **Ar→Pr**, **Ar→Rw**, **Cl→Ar**, **Cl→Pr**, **Cl→Rw**, **Pr→Ar**, **Pr→Cl**, **Pr→Rw**, **Rw→Ar**, **Rw→Cl**, and **Rw→Pr**.

The VisDA-2017 dataset is a challenging simulation-to-real dataset with over 280K images across 12 classes. It contains two distinct domains: *Synthetic* (**S**) that has renderings of 3D models from different angles under different lighting conditions, and *Real* (**R**) that contains natural images. Following [15], we select the first 6 categories of each domain in the alphabetical order as the target classes and obtain two PDA tasks: **R→S** and **S→R**.

We compare the SPDA method with state-of-the-art DA and PDA methods, including Deep Adaptation Network (DAN) [17], Domain Adversarial Neural Network (DANN) [7], Adversarial Discriminative Domain Adaptation (ADDA) [31], Partial Adversarial Domain Adaptation (PADA) [2], Selective Adversarial Network (SAN) [1], Importance Weighted Adversarial Network (IWAN) [37], Example Transfer Network (ETN) [3], Deep Residual Correction Network (DRCN) [15], Reinforced Transfer Network (RTNet) [4], BA$^3$US [16], Selective Representation Learning for Class-Weight Computation (SRLCWC) [5], and Domain Consensus Clustering (DCC) [14]. We also compare with the ResNet-50 which is trained on the source samples only. Results of most baseline methods are directly from previous papers, including ETN [3], RTNet [4], BA$^3$US [16], DRCN [15], SRLCWC [5], and DCC [14]. Experimental results shown in italics indicate that we run public source code to obtain the results.

The ResNet-50 [11] pre-trained on ImageNet [25] is used as the backbone. After the backbone, we add new layers, which consist of a bottleneck block and a classification layer. The bottleneck block consists of a fully connected layer and a batch normalization layer with ReLU activation functions as well as the dropout operation. The classification layer is a fully connected layer. These new layers are trained from scratch and their learning rates are 10 times that of the backbone that will be fine-tuned. For optimization, we adopt the mini-batch SGD with the Nesterov momentum 0.9. The learning rate is adjusted by $\eta_t = \frac{\eta_0}{(1+\alpha t)^\beta}$, where $t$ denotes the training step, $\alpha = 0.001$, $\beta = 0.75$, and $\eta_0 = 0.1$ for new layers. In Eq. (5), we set $\lambda = \frac{2}{1+\exp((-10*p)/P))} - 1$, where $p$ is the index of current training epoch and $P$ is the total number of training epochs. Note that an increasing $\lambda$ helps training a better model. Specifically, at the beginning of the training process, the extracted features are not very good and so the weight of MoC should not be large. As the training process proceeds, the network can extract better features, thus the weight of MoC could be increasing. Furthermore, we set the threshold $\theta$ in Eq. (3) to 0.9. The batch size is set to 128 for all the datasets. We report the average classification accuracy and standard deviation over 3 random trials. We implement all the methods based on the PyTorch package [21].

## A.2 Why Selective Training?

In this section, we explain why we use the selective training method instead of the original self-training strategy that only assigns pseudo-labels to the unlabeled target data and computes the classification loss with these data. Hence, the selected target samples in the self-training strategy will not be used to compute the MoC similarity. Specifically, the objective function of the SPDA method with the self-training strategy is formulated as

$$\min_{\mathbf{w}} \mathcal{L}_C(\widetilde{X}_S, \{y_s, \hat{y}_t\}) - \lambda \mathrm{MoC}(X_S, X_T), \tag{9}$$

which differs from problem (5) in that $X_S$ instead of $\widetilde{X}_S$ is included in the calculation of the MoC similarity.

We compare the proposed SPDA method with problem (9) on three hard transfer tasks (i.e., Ar→Cl, Pr→Cl, and Rw→Cl) on the Office-Home dataset. According to experimental results shown in Table 6, where the SPDA-SelfT method corresponds to problem (9), we can see that the selective training method outperforms the self-training strategy in all the transfer tasks. One possible reason is that after adding these target samples to the source domain, the MoC similarity selects not only source samples but also some target samples, which make the proposed SPDA method not only improve the inter-domain similarity but also the intra-domain similarity.

| Method | Ar→Cl | Pr→Cl | Rw→Cl | Avg |
|---|---|---|---|---|
| SPDA-SelfT | 58.93 | 58.87 | 66.45 | 61.42 |
| SPDA | 64.24 | 58.91 | 67.41 | 63.52 |

Table 6: Accuracy (%) of the SPDA and SPDA-SelfT methods corresponding to problems (5) and (9), respectively, on three transfer tasks of the Office-Home dataset.

## A.3 Visualization

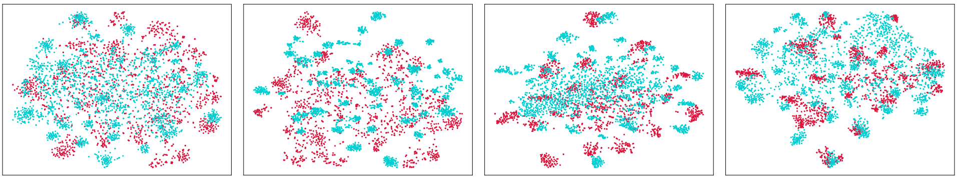
(a) ResNet-50 (b) DANN (c) BA$^3$US (d) SPDA (Ours)

Figure 5: t-SNE visualizations for the transfer task Ar→Cl on the Office-Home dataset. The cyan points indicate source samples and the red points represent target samples.

We visualize in Fig. 5 t-SNE embeddings [6] of hidden features learned by ResNet-50, DANN, BA$^3$US, and SPDA on the transfer task Ar→Cl of the Office-Home dataset. According to Fig. 5, we can see that the proposed SPDA is more discriminative on target data (i.e., red points) and can effectively match target classes to the relevant source classes than other methods in this task, which is a difficult task as the state-of-the-art performance just achieved by the proposed SPDA method is only 64.24% in terms of the accuracy. Specifically, the feature representations of the target domain learned by ResNet-50 and DANN are mixed across classes, which implies that ResNet-50 and DANN cannot discriminate target data very well. Furthermore, the feature representation learned by the proposed SPDA method is clustered more clearly than BA$^3$US, which indicates that SPDA can better discriminate target examples than BA$^3$US.