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 λ 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 θ 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

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

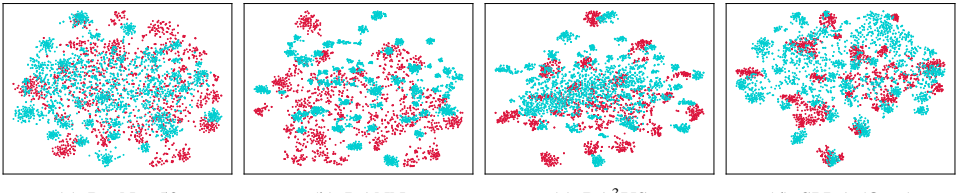
which differs from problem (5) in that X_S instead of $\widetilde{X}_S$ is included in the calculation of the MoC similarity.

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.