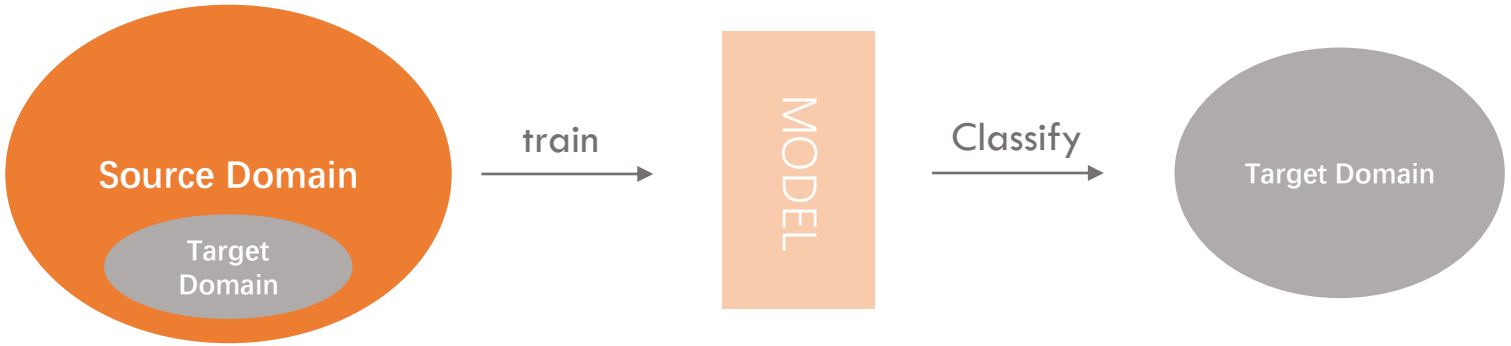
Unsupervised Domain Adaptation Via Structured Prediction Based Selective Pseudo-Labeling

Qian Wang,[1] **Toby P. Breckon**[1,2]
[1]Department of Computer Science, Durham University, United Kingdom
[2]Department of Engineering, Durham University, United Kingdom
qian.wang173@hotmail.com, toby.breckon@durham.ac.uk

AAAI - 2020

INTRODUCTION

- UDA (Unsupervised Domain Adaptation) aims to address the problem of classifying unlabeled samples from the target domain whilst labeled samples are only available from the source domain and the data distributions are different in these two domains.



- Approaches to UDA have been proposed trying to align the marginal distributions of source and target domain which is not guaranteed to produce good classification results as the conditional distribution of the target domain can be misaligned with that of the source domain.

- Pseudo labeling the target samples allows to align the conditional distributions of source and target domains with traditional supervised learning algorithms.

Approaches without Pseudo-Labeling

- Maximum Mean Discrepancy (MMD)
- GRL (Gradient reversal layer)
- Generative Adversarial Loss

Pseudo-Labeling without Selection

Pseudo-labeling without selection assigns pseudo-labels to all samples in the target domain.

- The strategy of hard labeling assigns a pseudo-label y^ to each unlabeled sample without considering the confidence.
- The strategy of soft labeling assigns the conditional probability of each class p (c | x) given a target sample x which results in a pseudo-labeling vector.

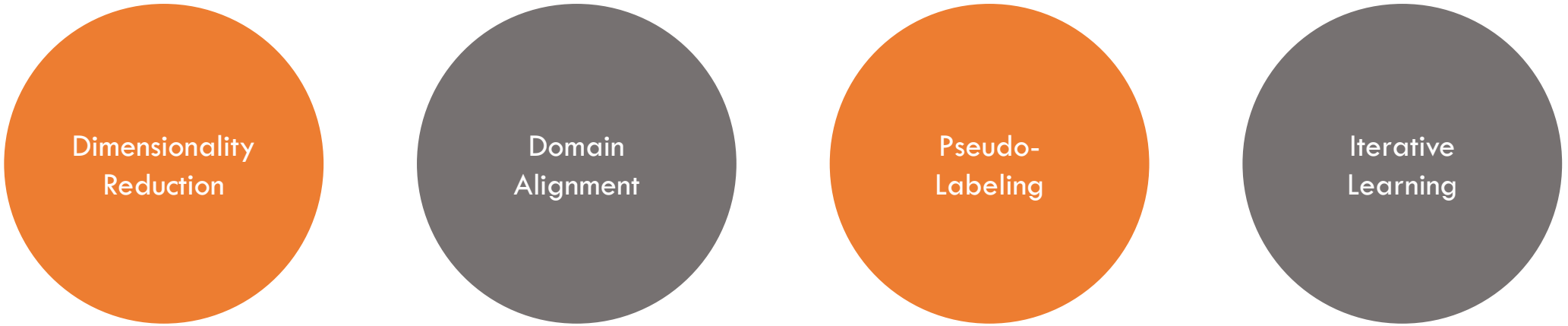
Pseudo-Labeling with Selection

A subset of target samples are selected to be assigned with pseudo labels and only these pseudo-labeled target samples are combined with source samples in the next iteration of learning. One key factor in such algorithms is the criterion of sample selection for pseudo-labeling.

- class-wise sample selection strategy : Samples are selected for each class independently.

METHODS

The proposed method aims to align the **conditional distributions** of source and target domains.



- High dimensional features contain redundant information and thus result in unnecessary computation.

$$\mathbf{X} = [\boldsymbol{x}_1^s, ..., \boldsymbol{x}_{n_s}^s, \boldsymbol{x}_1^t, ..., \boldsymbol{x}_{n_t}^t] \in \mathbb{R}^{d \times n}$$

$$\mathbf{X}\mathbf{H}\mathbf{X}^T\boldsymbol{v} = \phi\boldsymbol{v}. \quad (2) \quad \mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$$

$$\tilde{\mathbf{X}} = \mathbf{V}^T\mathbf{X} \quad (3) \quad \mathbf{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_{d_1}] \in \mathbb{R}^{d \times d_1}$$

$$\tilde{\mathbf{X}} \in \mathbb{R}^{d_1 \times n} \quad d_1 \leq d$$

- L2 normalization is applied to each feature vector in $\tilde{\mathbf{X}}$ as $\tilde{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{x}}/||\tilde{\boldsymbol{x}}||_2$

The use of L2 normalization forces samples of both source and target domains distributed on the surface of the same hyper-sphere which helps to align data from different domains.

Learning a projection matrix P with SLPP (Supervised Locality Preserving Projection) which maps samples from both domains into the same latent subspace Z from X.

$$\min_{\mathbf{P}} \sum_{i,j} ||\mathbf{P}^T \tilde{\boldsymbol{x}}_i - \mathbf{P}^T \tilde{\boldsymbol{x}}_j||_2^2 \mathbf{M}_{ij}, \qquad (4)$$

$\mathbf{P} \in \mathbb{R}^{d_1 \times d_2}$ $\quad \tilde{\mathbf{X}}^l \in \mathbb{R}^{d_1 \times (n_s + n_t')}$ $\quad \mathbf{M} \in \mathbb{R}^{(n_s+n_t') \times (n_s+n_t')}$

$$\mathbf{M}_{ij} = \begin{cases} 1, & y_i = y_j, \\ 0, & otherwise. \end{cases} \qquad (5)$$

[1] $$\max_{\mathbf{P}} \frac{tr(\mathbf{P}^T \tilde{\mathbf{X}}^l \mathbf{D} \tilde{\mathbf{X}}^{lT} \mathbf{P})}{tr(\mathbf{P}^T (\tilde{\mathbf{X}}^l \mathbf{L} \tilde{\mathbf{X}}^{lT} + \mathbf{I}) \mathbf{P})} \qquad (6)$$

$\mathbf{L} = \mathbf{D} - \mathbf{M}$

$\mathbf{D}_{ii} = \sum_j \mathbf{M}_{ij}$

$tr(\mathbf{P}^T \mathbf{P})$ is added for penalizing extreme values in the projection matrix P

The idea is that samples from the same class should be projected close to each other in the subspace regardless of which domain they are originally from.

[1] He, X., and Niyogi, P . 2004. Locality preserving projections. In Advances in neural information processing systems, 153–160.

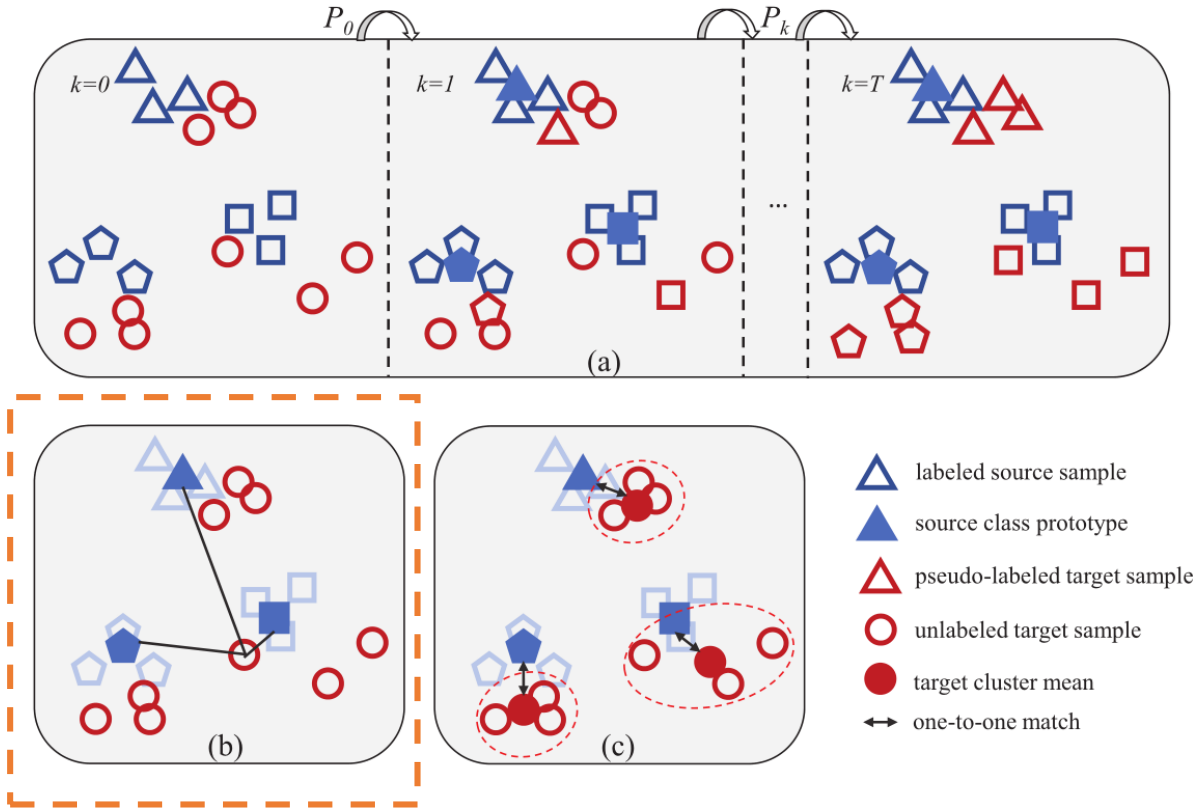
- Pseudo-Labeling via nearest Class Prototype (NCP)

$$\boldsymbol{z}^s = \mathbf{P}^T \tilde{\boldsymbol{x}}^s, \quad \boldsymbol{z}^t = \mathbf{P}^T \tilde{\boldsymbol{x}}^t. \tag{8}$$

$$\bar{\boldsymbol{z}}_y^s = \frac{\sum_{i=1}^{n_s} \boldsymbol{z}_i^s \delta(y, y_i^s)}{\sum_{i=1}^{n_s} \delta(y, y_i^s)}, \tag{9}$$

$\delta(y, y_i) = 1$ if $y = y_i$ and 0 otherwise.

$$p_1(y|\boldsymbol{x}^t) = \frac{\exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^s||\right)}{\sum_{y=1}^{|\mathcal{Y}|} \exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^s||\right)}. \tag{10}$$



- Pseudo-Labeling via Structured Prediction(SP)

NCP does not consider the intrinsic structure of the target samples which provides useful information for target samples classification.

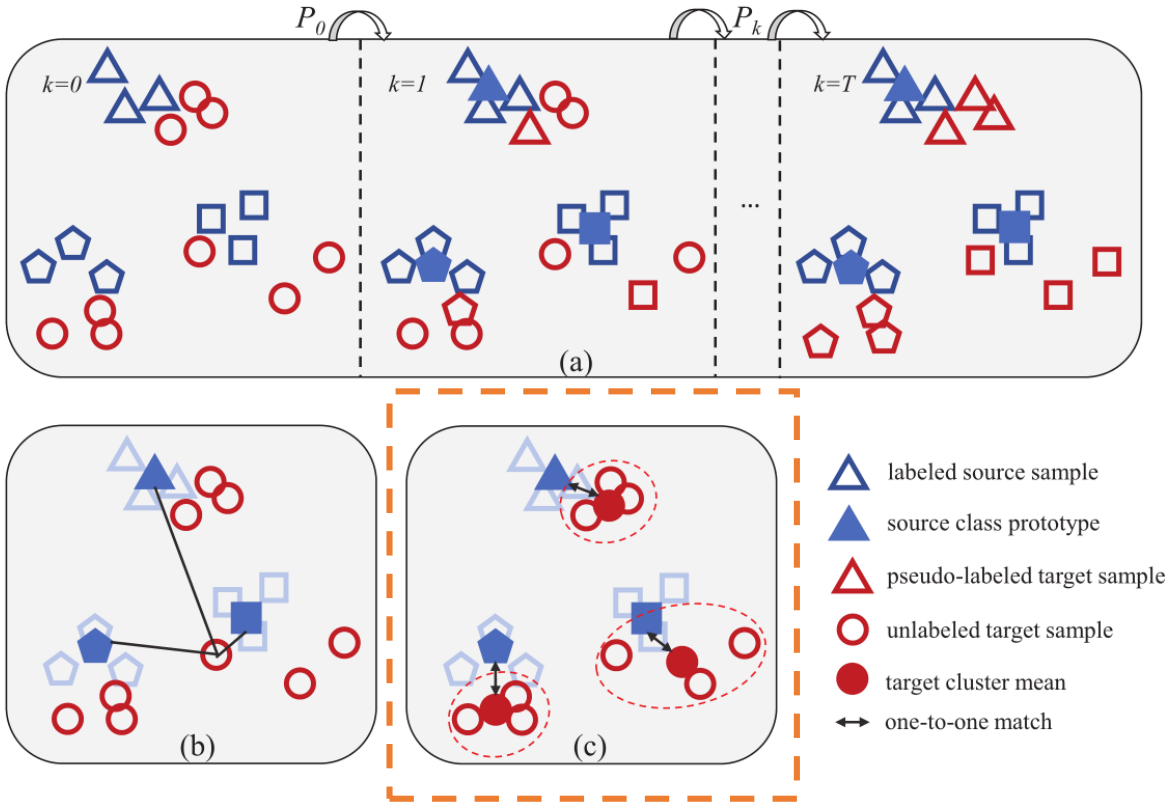
One-to-one match between a **cluster** from the target domain and a **class** from the source domain so that the sum of distances of all the matched **pairs** of the cluster center and the class prototype is minimised.

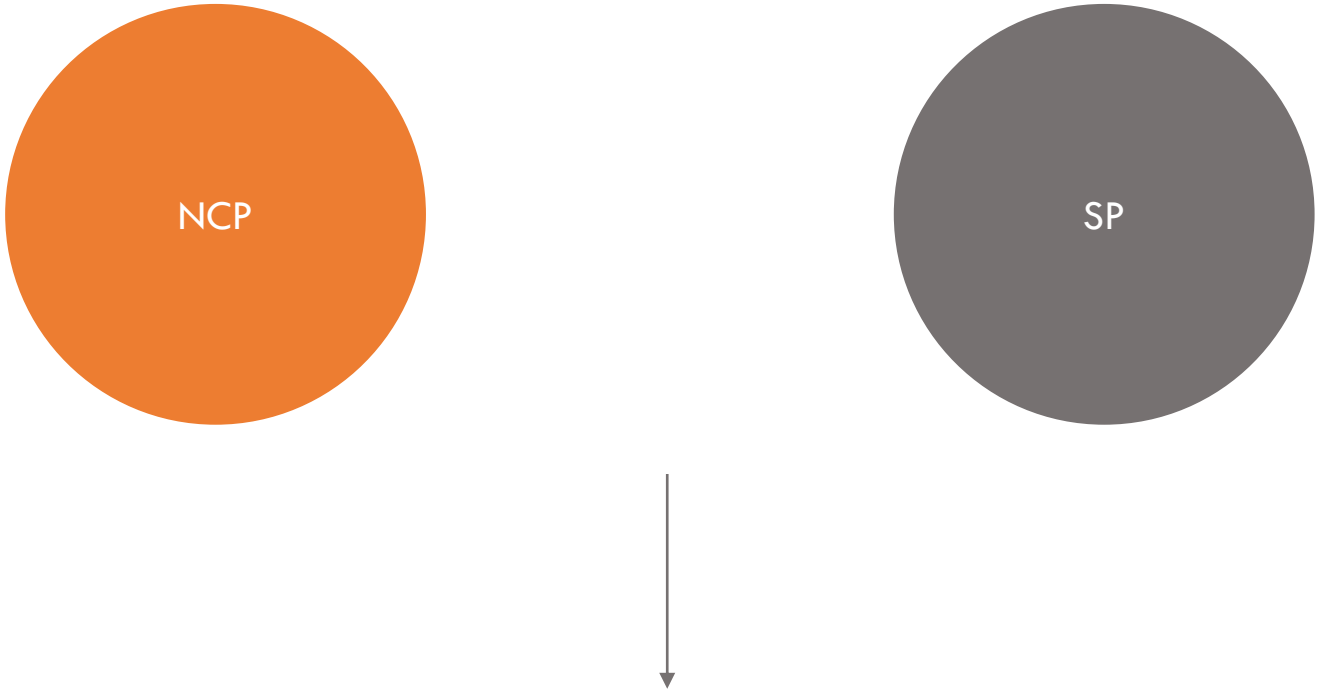
$$\min_{\mathbf{A}} \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=1}^{|\mathcal{Y}|} \mathbf{A}_{ij} d(\bar{\boldsymbol{z}}_i^t, \bar{\boldsymbol{z}}_j^s)$$
$$s.t. \quad \forall i, \sum_j \mathbf{A}_{ij} = 1; \forall j, \sum_i \mathbf{A}_{ij} = 1, \tag{11}$$

$\bar{\boldsymbol{z}}_i^t$ denotes the i-th cluster center in the target domain.

$\mathbf{A} \in \{0,1\}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ denote the one-to-one matching matrix.

$$p_2(y|\boldsymbol{x}^t) = \frac{\exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^t||\right)}{\sum_{y=1}^{|\mathcal{Y}|} \exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^t||\right)}. \tag{12}$$





$$p(y|\boldsymbol{x}^t) = \max\{p_1(y|\boldsymbol{x}^t), p_2(y|\boldsymbol{x}^t)\}. \tag{13}$$

$$\hat{y}^t = \arg\max_{y \in \mathcal{Y}} p(y|\boldsymbol{x}^t). \tag{14}$$

An iterative learning strategy is used to learn the projection matrix P for domain alignment and improved pseudo-labeling for target samples alternately.

Algorithm 1 Unsupervised Domain Adaptation Using Selective Pseudo-Labeling

Input: Labeled source data set $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1, 2, ..., n_s$ and unlabeled target data set $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}, i = 1, 2, ..., n_t$, dimensionality of PCA and SLPP subspace d_1 and d_2, number of iteration T.

Output: The projection matrix $\mathbf{P}$ and predicted labels $\{\hat{y}^t\}$ for target samples.

1: Initialize $k = 0$;
2: Dimensionality reduction by Eq. (3);
3: Learn the projection $\mathbf{P}_0$ using only source data $\mathcal{D}^s$;
4: Assign pseudo labels for all target data using Eq. (14);
5: **while** $k < T$ **do**
6: $k \leftarrow k + 1$;
7: Select a subset of pseudo-labeled target data $\mathcal{S}_k \in \hat{\mathcal{D}}^t$;
8: Learn P_k using $\mathcal{D}^s$ and $\mathcal{S}_k$;
9: Update pseudo labels for all target data using Eq.(14).
10: **end while**

For each class, $c \in \mathcal{Y}$ Pick out n_t^c target samples pseudo-labeled as class c from

Which we select top kn_t^c/T high-probability samples to form $\mathcal{S}_k$

$\mathcal{S}_k \subseteq \hat{\mathcal{D}}^t$ containing kn_t/T arget samples in the k-th iteration

Dataset : Office-Caltech

Source ⟶ Target

Table 1: Classification Accuracy (%) on Office-Caltech dataset using Decaf6 features. Each column displays the results of a pair of source → target setting.

Method	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
DDC(Tzeng et al. 2014)	91.9	85.4	88.8	85.0	86.1	89.0	78.0	84.9	**100.0**	81.1	89.5	98.2	88.2
DAN(Long et al. 2015)	92.0	90.6	89.3	84.1	91.8	91.7	81.2	92.1	**100.0**	80.3	90.0	98.5	90.1
DCORAL(Sun and Saenko 2016)	92.4	91.1	91.4	84.7	-	-	79.3	-	-	82.8	-	-	-
CORAL(Sun, Feng, and Saenko 2017)	92.0	80.0	84.7	83.2	74.6	84.1	75.5	81.2	**100.0**	76.8	85.5	99.3	84.7
SCA(Ghifary et al. 2016)	89.5	85.4	87.9	78.8	75.9	85.4	74.8	86.1	**100.0**	78.1	90.0	98.6	85.9
JGSA(Zhang, Li, and Ogunbona 2017)	91.4	86.8	93.6	84.9	81.0	88.5	85.0	90.7	**100.0**	86.2	92.0	99.7	90.0
MEDA(Wang et al. 2018)	**93.4**	**95.6**	91.1	**87.4**	88.1	88.1	**93.2**	**99.4**	99.4	87.5	**93.2**	97.6	92.8
CAPLS (Wang, Bu, and Breckon 2019)	90.8	85.4	95.5	86.1	87.1	**94.9**	88.2	92.3	**100.0**	**88.8**	93.0	**100.0**	91.8
SPL (Ours)	92.7	93.2	**98.7**	**87.4**	**95.3**	89.2	87.0	92.0	**100.0**	88.6	92.9	98.6	**93.0**

Dataset : Office31 (left) ImageCLEF-DA (right)

Source ⟶ Target

Table 2: Classification Accuracy (%) on Office31 dataset using either ResNet50 features or ResNet50 based deep models.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
RTN(Long et al. 2016)	84.5	96.8	99.4	77.5	66.2	64.8	81.6
MADA(Pei et al. 2018)	90.0	97.4	99.6	87.8	70.3	66.4	85.2
GTA (Sankaranarayanan et al. 2018)	89.5	97.9	99.8	87.7	72.8	71.4	86.5
iCAN(Zhang et al. 2018)	92.5	**98.8**	**100.0**	90.1	72.1	69.9	87.2
CDAN-E(Long et al. 2018)	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
JDDA(Chen et al. 2019a)	82.6	95.2	99.7	79.8	57.4	66.7	80.2
SymNets(Zhang et al. 2019)	90.8	**98.8**	**100.0**	**93.9**	74.6	72.5	88.4
TADA (Wang et al. 2019)	**94.3**	98.7	99.8	91.6	72.9	73.0	88.4
MEDA(Wang et al. 2018)	86.2	97.2	99.4	85.3	72.4	74.0	85.7
CAPLS (Wang, Bu, and Breckon 2019)	90.6	98.6	99.6	88.6	75.4	76.3	88.2
SPL (Ours)	92.7	98.7	99.8	93.0	**76.4**	**76.8**	**89.6**

Table 3: Classification Accuracy (%) on ImageCLEF-DA dataset using either ResNet50 features or ResNet50 based deep models.

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
RTN(Long et al. 2016)	75.6	86.8	95.3	86.9	72.7	92.2	84.9
MADA(Pei et al. 2018)	75.0	87.9	96.0	88.8	75.2	92.2	85.8
iCAN(Zhang et al. 2018)	79.5	89.7	94.7	89.9	78.5	92.0	87.4
CDAN-E(Long et al. 2018)	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7
SymNets(Zhang et al. 2019)	**80.2**	93.6	97.0	93.4	78.7	**96.4**	89.9
MEDA(Wang et al. 2018)	79.7	92.5	95.7	92.2	78.5	95.5	89.0
SPL (Ours)	78.3	**94.5**	96.7	**95.7**	**80.5**	96.3	**90.3**

Dataset : Office-Home

Source ⟶ Target

Table 4: Classification Accuracy (%) on Office-Home dataset using either ResNet50 features or ResNet50 based deep models.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Average
JAN(Long et al. 2017)	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN-E (Long et al. 2018)	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
SymNets (Zhang et al. 2019)	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	**74.5**	52.6	82.7	67.6
TADA (Wang et al. 2019)	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
MEDA(Wang et al. 2018)	54.6	75.2	77.0	56.5	72.8	72.3	59.0	51.9	78.2	67.7	57.2	81.8	67.0
CAPLS (Wang, Bu, and Breckon 2019)	**56.2**	**78.3**	80.2	**66.0**	75.4	78.4	**66.4**	**53.2**	81.1	71.6	56.1	84.3	70.6
SPL (Ours)	54.5	77.8	**81.9**	65.1	**78.0**	**81.1**	66.0	53.1	**82.8**	69.9	55.3	**86.0**	**71.0**

Ablation Study

Table 5: Results of ablation study.

Method				Office-Caltech	Office31	ImageCLEF-DA	Office-Home
PL	S	NCP	SP				
✗	✗	✓	✗	81.8	82.0	86.2	63.9
✗	✗	✗	✓	90.3	87.5	89.5	68.0
✗	✗	✓	✓	90.7	87.6	89.4	68.1
✓	✗	✓	✗	85.5	83.7	86.9	66.2
✓	✗	✗	✓	91.9	88.0	90.0	68.9
✓	✗	✓	✓	92.0	88.0	90.0	69.0
✓	✓	✓	✗	90.8	87.8	89.0	70.8
✓	✓	✗	✓	93.0	89.5	90.2	71.0
✓	✓	✓	✓	93.0	89.6	90.3	71.0

PL : Pseudo-Labeling

S : Selection

NCP : nearest Class Prototype

SP : Structured Prediction

Thanks