Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation

Xiangyu Yue[1,*] Zangwei Zheng[2,*] Shanghang Zhang[1] Yang Gao[3]
Trevor Darrell[1] Kurt Keutzer[1] Alberto Sangiovanni Vincentelli[1]
[1]UC Berkeley [2]Nanjing University [3]Tsinghua University

2021-10-20 Yujeong Lee

1 Methods

CVPR 2021
(https://openaccess.thecvf.com/content/CVPR2021/papers/Yue_Prototypical_Cross-Domain_Self-Supervised_Learning_for_Few-Shot_Unsupervised_Domain_Adaptation_CVPR_2021_paper.pdf)

Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation

(1) In-domain Prototypical Contrastive Learning

$$\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s} \quad \mathcal{D}_{su} = \{(\mathbf{x}_i^{su})\}_{i=1}^{N_{su}} \quad \mathcal{D}_{tu} = \{(\mathbf{x}_i^{tu})\}_{i=1}^{N_{tu}}$$

Stored vectors, V

feature encoder F.

feature vector $\mathbf{f}_i^s = F(\mathbf{x}_i^s)$

$$\mathbf{v}_i \leftarrow m\mathbf{v}_i + (1-m)\mathbf{f}_i$$

$\mathbf{v}_i$ is the stored feature vector of $\mathbf{x}_i$

$$\boldsymbol{V}^s = [\mathbf{v}_1^s, \cdots, \mathbf{v}_{(N_s+N_{su})}^s], \ \boldsymbol{V}^t = [\mathbf{v}_1^t, \cdots, \mathbf{v}_{N_{tu}}^t]$$

Normalized prototypes

$$\mu_j^s = \frac{\mathbf{u}_j^s}{\|\mathbf{u}_j^s\|} \qquad \mathbf{u}_j^s = \frac{1}{|C_j^{(s)}|}\sum_{\mathbf{v}_i^s \in C_j^{(s)}} \mathbf{v}_i^s$$

normalized source prototypes $\{\mu_j^s\}_{j=1}^k$

target prototypes $\{\mu_j^t\}_{j=1}^k$

Get cluster by k-means clustering

$$\boldsymbol{C}^s = \{C_1^{(s)}, C_2^{(s)}, \ldots, C_k^{(s)}\}$$

$$P_i^s = [P_{i,1}^s, P_{i,2}^s, \ldots, P_{i,k}^s]$$

$$P_{i,j}^s = \frac{\exp(\mu_j^s \cdot \mathbf{f}_i^s/\phi)}{\sum_{r=1}^k \exp(\mu_r^s \cdot \mathbf{f}_i^s/\phi)}$$

$$\mathcal{L}_{\mathrm{PC}} = \sum_{i=1}^{N_s+N_{su}} \mathcal{L}_{CE}(P_i^s, c_s(i)) + \sum_{i=1}^{N_{tu}} \mathcal{L}_{CE}(P_i^t, c_t(i))$$

where $c_s(\cdot)$ and $c_t(\cdot)$ return the cluster index of the instance

Perform k-means on the samples M times with different number of cluster

$$\mathcal{L}_{\mathrm{InSelf}} = \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}_{\mathrm{PC}}^{(m)} \tag{5}$$

CVPR 2021
(https://openaccess.thecvf.com/content/CVPR2021/papers/Yue_Prototypical_Cross-Domain_Self-Supervised_Learning_for_Few-Shot_Unsupervised_Domain_Adaptation_CVPR_2021_paper.pdf)

Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation

(2) Cross-domain Instance-Prototype SSL

$$P_{i,j}^{s\to t} = \frac{\exp(\mu_j^t \cdot \mathbf{f}_i^s/\tau)}{\sum_{r=1}^{k}\exp(\mu_r^t \cdot \mathbf{f}_i^s/\tau)}$$

$$\mathcal{L}_{\text{CrossSelf}} = \sum_{i=1}^{N_s+N_{su}} H(P_i^{s\to t}) + \sum_{i=1}^{N_{tu}} H(P_i^{t\to s})$$

(3) Adaptive Prototypical Classifier Learning

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_s}\mathcal{L}_{CE}(\mathbf{p}(\mathbf{x}), y)$$

Prototype Classifier Update

cosine classifier C consists of weight vectors $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n_c}]$

$$\mathbf{p}(\mathbf{x}) = \sigma(\tfrac{1}{T}\mathbf{W}^{\mathrm{T}}\mathbf{f}) \quad \mathbf{p}(\mathbf{x}) = [\mathbf{p}(\mathbf{x})_1, \ldots, \mathbf{p}(\mathbf{x})_{n_c}]$$

$$\mathcal{D}_s^{(i)} = \{\mathbf{x}|(\mathbf{x}, y) \in \mathcal{D}_s, y = i\} \quad \mathcal{D}_{su}^{(i)} = \{\mathbf{x}|\mathbf{x} \in \mathcal{D}_{su}, \mathbf{p}(\mathbf{x})_i > t\}$$

$$\hat{\mathbf{w}}_i^s = \frac{1}{|\mathcal{D}_{s+}^{(i)}|}\sum_{\mathbf{x}\in\mathcal{D}_{s+}^{(i)}} \boldsymbol{V}^s(\mathbf{x}) \quad \hat{\mathbf{w}}_i^t = \frac{1}{|\mathcal{D}_{tu}^{(i)}|}\sum_{\mathbf{x}\in\mathcal{D}_{tu}^{(i)}} \boldsymbol{V}^t(\mathbf{x})$$

$$\mathcal{D}_{s+}^{(i)} = \mathcal{D}_s^{(i)} \cup \mathcal{D}_{su}^{(i)}$$

$$\mathbf{w}_i = \begin{cases} unit(\hat{\mathbf{w}}_i^s) & \text{if } |\mathcal{D}_{tu}^{(i)}| < t_w \\ unit(\hat{\mathbf{w}}_i^t) & \text{otherwise} \end{cases}$$

Mutual Information Maximization

(1) To promote the network to have diversified outputs over the dataset
→ Maximize the entropy of expected network prediction

(2) To get high confident prediction for each sample
→ entropy minimization on the network output

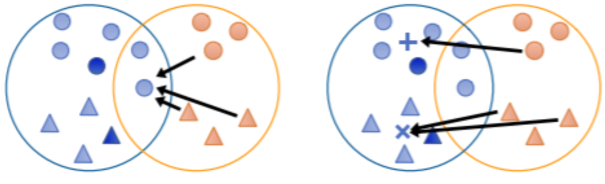
$$\mathcal{I}(y;\mathbf{x}) = \mathcal{H}(\mathbf{p}_0) - \mathbb{E}_{\mathbf{x}}[\mathcal{H}(p(y|\mathbf{x};\theta))]$$

$$\mathcal{L}_{\text{MIM}} = -\mathcal{I}(y;\mathbf{x})$$

Total Loss

CVPR 2021
(https://openaccess.thecvf.com/content/CVPR2021/papers/Yue_Prototypical_Cross-Domain_Self-Supervised_Learning_for_Few-Shot_Unsupervised_Domain_Adaptation_CVPR_2021_paper.pdf)

Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation



$$\mathcal{L}_{PCS} = \mathcal{L}_{\text{cls}} + \lambda_{\text{in}} \cdot \mathcal{L}_{\text{InSelf}} + \lambda_{\text{cross}} \cdot \mathcal{L}_{\text{CrossSelf}} + \lambda_{\text{mim}} \cdot \mathcal{L}_{\text{MIM}}$$

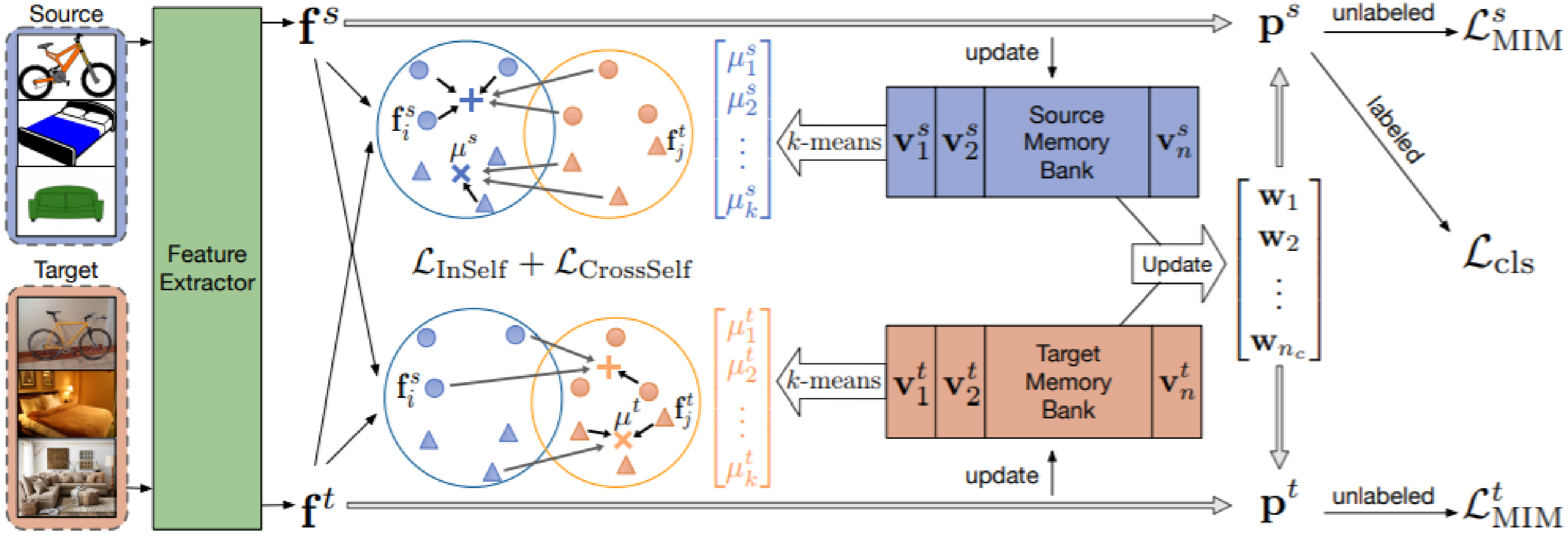


Figure 2: An overview of the PCS framework. In-domain and cross-domain self-supervision are performed between normalized feature vectors $\mathbf{f}$ and prototypes μ computed by clustering vectors $\mathbf{v}$ in memory banks. Features with confident predictions ($\mathbf{p}$) are used to adaptively update classifier vectors $\mathbf{w}$. MI maximization and classification loss are further used to extract discriminative features.

3 Results

CVPR 2021
(https://openaccess.thecvf.com/content/CVPR2021/papers/Yue_Prototypical_Cross-Domain_Self-Supervised_Learning_for_Few-Shot_Unsupervised_Domain_Adaptation_CVPR_2021_paper.pdf)

Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation

Table 1: Adaptation accuracy (%) comparison on 1-shot and 3-shots per class on the Office dataset.

Method	Office: Target Acc. on 1-shot / 3-shots						
	A→D	A→W	D→A	D→W	W→A	W→D	Avg
SO	27.5 / 49.2	28.7 / 46.3	40.9 / 55.3	65.2 / 85.5	41.1 / 53.8	62.0 / 86.1	44.2 / 62.7
MME [59]	21.5 / 51.0	12.2 / 54.6	23.1 / 60.2	60.9 / 89.7	14.0 / 52.3	62.4 / 91.4	32.3 / 66.5
CDAN [45]	11.2 / 43.7	6.2 / 50.1	9.1 / 65.1	54.8 / 91.6	10.4 / 57.0	41.6 / 89.8	22.2 / 66.2
SPL [71]	12.0 / 77.1	7.7 / 80.3	7.3 / 74.2	7.2 / 93.5	7.2 / 64.4	10.2 / 91.6	8.6 / 80.1
CAN [38]	25.3 / 48.6	26.4 / 45.3	23.9 / 41.2	69.4 / 78.2	21.2 / 39.3	67.3 / 82.3	38.9 / 55.8
MDDIA [35]	45.0 / 62.9	54.5 / 65.4	55.6 / 67.9	84.4 / 93.3	53.4 / 70.3	79.5 / 93.2	62.1 / 75.5
CDS [39]	33.3 / 57.0	35.2 / 58.6	52.0 / 67.6	59.0 / 86.0	46.5 / 65.7	57.4 / 81.3	47.2 / 69.3
DANN + ENT [18]	32.5 / 57.6	37.2 / 54.1	36.9 / 54.1	70.1 / 87.4	43.0 / 51.4	58.8 / 89.4	46.4 / 65.7
MME + ENT	37.6 / 69.5	42.5 / 68.3	48.6 / 66.7	73.5 / 89.8	47.2 / 63.2	62.4 / 95.4	52.0 / 74.1
CDAN + ENT	31.5 / 68.3	26.4 / 71.8	39.1 / 57.3	70.4 / 88.2	37.5 / 61.5	61.9 / 93.8	44.5 / 73.5
CDS + ENT	40.4 / 61.2	44.7 / 66.7	66.4 / 73.1	71.6 / 90.6	58.6 / 71.8	69.3 / 86.1	58.5 / 74.9
CDS + MME + ENT	39.4 / 61.6	43.6 / 66.3	66.0 / 74.5	75.7 / 92.1	53.1 / 73.0	70.9 / 90.6	58.5 / 76.3
CDS + CDAN + ENT	52.6 / 65.1	55.2 / 68.8	65.7 / 71.2	76.6 / 88.1	59.7 / 71.0	73.3 / 87.3	63.9 / 75.3
CDS / MME + ENT†	55.4 / 75.7	57.2 / 77.2	62.8 / 69.7	84.9 / 92.1	62.6 / 69.9	77.7 / 95.4	65.3 / 80.0
CDS / CDAN + ENT†	53.8 / 78.1	65.6 / 79.8	59.5 / 70.7	83.0 / 93.2	57.4 / 64.5	77.1 / 97.4	66.1 / 80.6
PCS (Ours)	**60.2 / 78.2**	**69.8 / 82.9**	**76.1 / 76.4**	**90.6 / 94.1**	**71.2 / 76.3**	**91.8** / 96.0	**76.6 / 84.0**
Improvement	+4.8 / +0.1	+4.2 / +3.1	+9.7 / +1.9	+5.7 / +0.9	+8.6 / +3.3	+14.1 / -1.4	+10.5 / +3.4

† Two-stage training results reported in [39].

3

Results

CVPR 2021
(https://openaccess.thecvf.com/content/CVPR2021/papers/Yue_Prototypical_Cross-Domain_Self-Supervised_Learning_for_Few-Shot_Unsupervised_Domain_Adaptation_CVPR_2021_paper.pdf)

Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation

Table 2: Performance contribution of each part in PCS framework on Office.

Method	Office: Target Acc. on 1-shot / 3-shots						
	A→D	A→W	D→A	D→W	W→A	W→D	Avg
$\mathcal{L}_{\text{cls}}$	27.5 / 49.2	28.7 / 46.3	40.9 / 55.3	65.2 / 85.5	41.1 / 53.8	62.0 / 86.1	44.2 / 62.7
$+\mathcal{L}_{\text{InSelf}}$	39.0 / 55.6	38.6 / 55.1	47.2 / 68.5	71.7 / 89.4	50.9 / 68.4	65.1 / 90.6	52.1 / 71.3
$+\mathcal{L}_{\text{CrossSelf}}$	47.2 / 71.1	52.7 / 70.6	59.0 / 75.5	76.4 / 90.3	58.5 / 74.1	66.9 / 91.8	60.1 / 78.9
$+\mathcal{L}_{\text{MIM}}$	52.8 / 73.5	57.5 / 71.2	67.2 / 76.3	78.9 / 91.4	64.2 / 74.3	68.7 / 92.2	64.9 / 79.8
+APCU (PCS)	**60.2 / 78.2**	**69.8 / 82.9**	76.1 / **76.4**	**90.6 / 94.1**	**71.2 / 76.3**	**91.8 / 96.0**	**76.6 / 84.0**
PCS w/o MIM	59.0 / 75.9	58.6 / 76.5	**76.2 / 76.4**	87.8 / 93.2	68.7 / 74.7	89.8 / 95.0	73.5 / 82.0

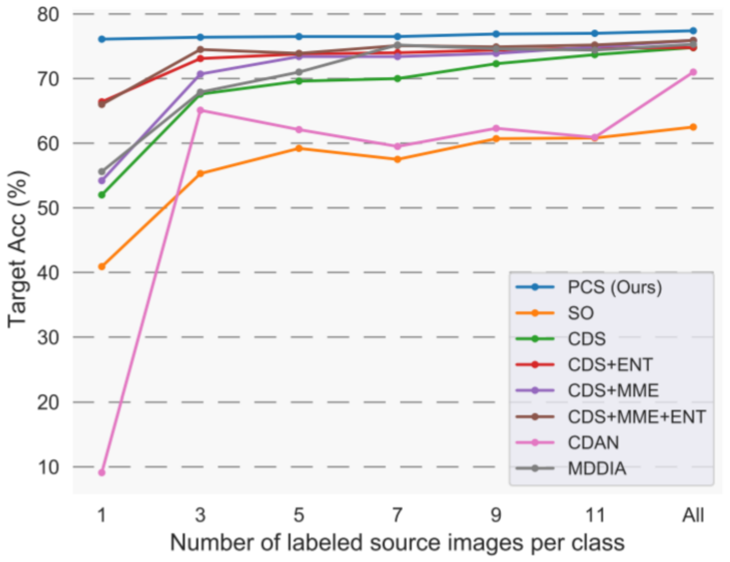


Figure 5: Sample efficiency comparison from DSLR to Amazon in Office dataset.

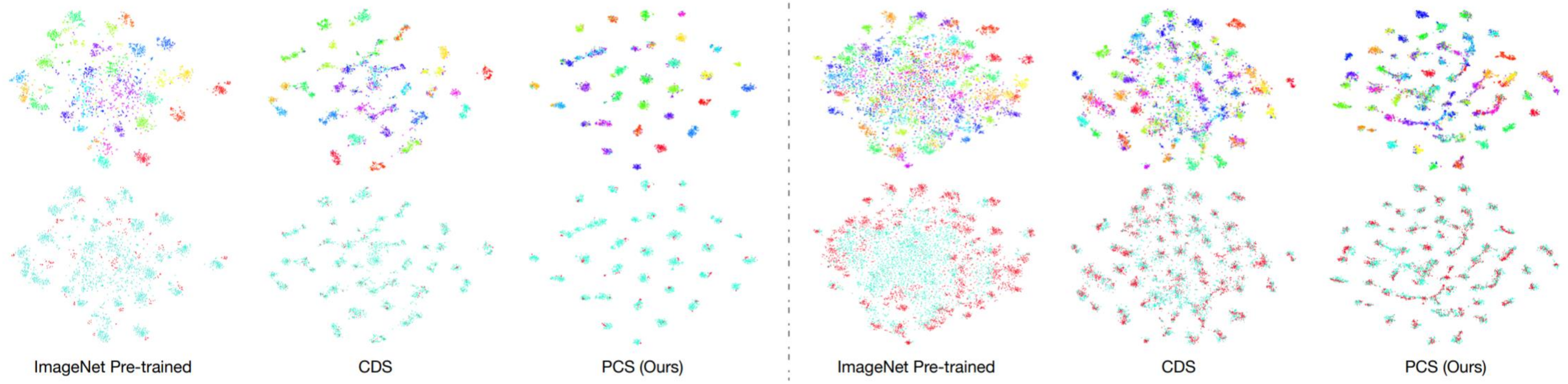


Figure 4: t-SNE visualization of ours and baselines on Office (left) and Office-Home (right). Top row: Coloring represents the class of each sample. Features with PCS are more discriminative than the ones with other methods. Bottom row: Cyan represents source features and Red represents target features. Feature from PCS are better-aligned between domains compared to other methods.

Top
Color : class of each sample

bottom
Cyan : source samples
Red : target samples