Self-Supervised Domain Adaptation with Consistency Training

Liang Xiao[1], Jiaolong Xu[1], Dawei Zhao[1], Zhiyu Wang[2], Li Wang[2], Yiming Nie[1] and Bin Dai[1,2]

[1]National Innovation Institute of Defense Technology, China (NIIDT)
[2]National University of Defense Technology, China (NUDT)

xiaoliang.cs@gmail.com, jiaolong@gmail.com

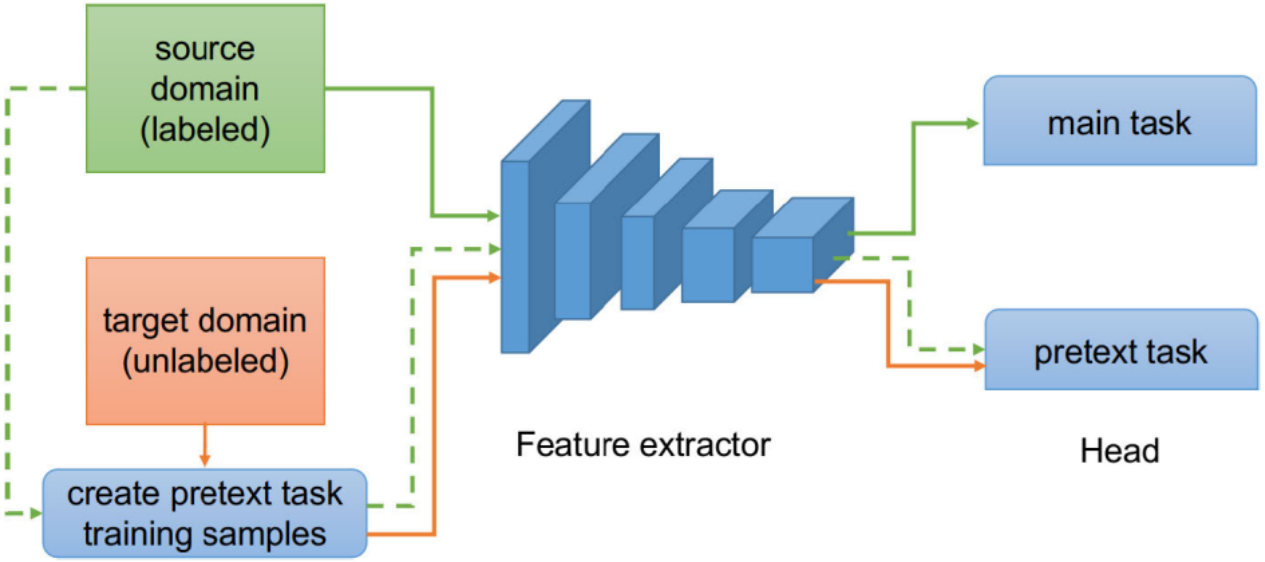
January 2021

Self-supervised Domain Adaptation [Xu, 2019]

$$\min_{\boldsymbol{\theta_e},\boldsymbol{\theta_m},\boldsymbol{\theta_p}} \mathcal{L}_m(\boldsymbol{\theta_e},\boldsymbol{\theta_m}) + \lambda_p \mathcal{L}_p(\boldsymbol{\theta_e},\boldsymbol{\theta_p}),$$

$$\mathcal{L}_m(\boldsymbol{\theta_e},\boldsymbol{\theta_m}) = \mathbb{E}_{\mathbf{x}^s, y^s \in D_s}\big[-\log p(y^s \mid \mathbf{x}^s)\big]$$

$$\mathcal{L}_p(\boldsymbol{\theta_e},\boldsymbol{\theta_p}) = \mathbb{E}_{\mathbf{x}^t \in D_t}\big[-\log p(\tilde{y}^t \mid \mathbf{x}^t)\big]$$

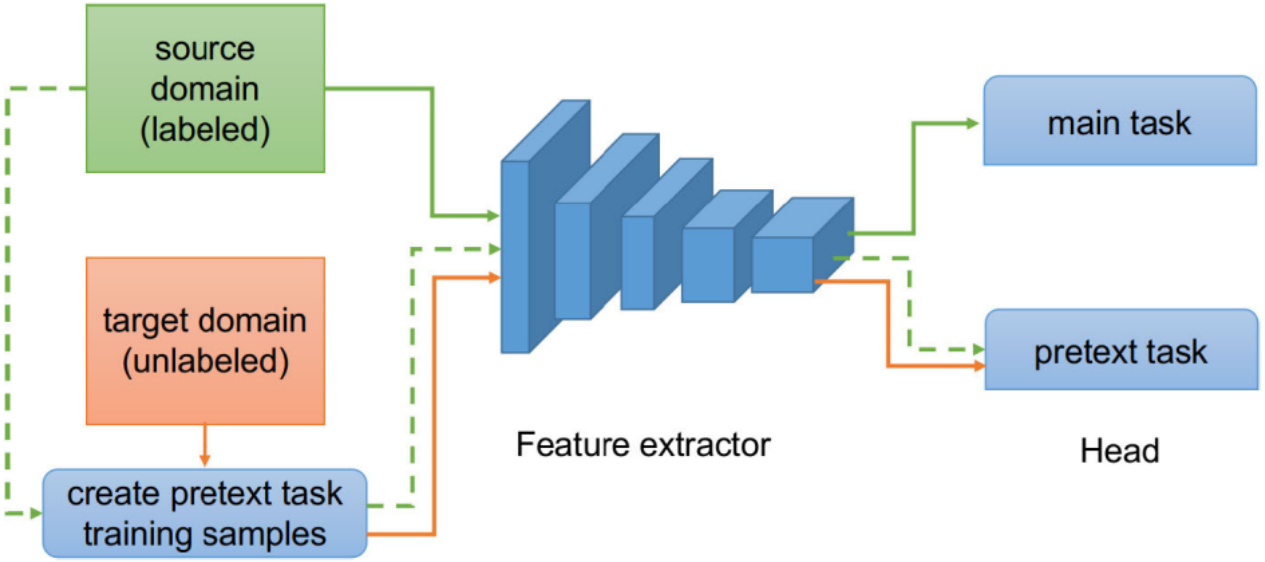


Self-supervised Domain Adaptation [Xu, 2019]

$$\min_{\boldsymbol{\theta_e},\boldsymbol{\theta_m},\boldsymbol{\theta_p}} \mathcal{L}_m(\boldsymbol{\theta_e},\boldsymbol{\theta_m}) + \lambda_p \mathcal{L}_p(\boldsymbol{\theta_e},\boldsymbol{\theta_p}),$$

$$\mathcal{L}_m(\boldsymbol{\theta_e},\boldsymbol{\theta_m}) = \mathbb{E}_{\mathbf{x}^s, y^s \in D_s}\big[-\log p(y^s \mid \mathbf{x}^s)\big]$$

$$\mathcal{L}_p(\boldsymbol{\theta_e},\boldsymbol{\theta_p}) = \mathbb{E}_{\mathbf{x}^t \in D_t}\big[-\log p(\tilde{y}^t \mid \mathbf{x}^t)\big]$$



Problem: No guarantee on the compatibility between the pretext task and the main task.

Self-supervised DA with Consistency Training

Explicitly relate the representation of the transformed (specifically, rotated) image to the label of the main task by maximizing the mutual information:

Assuming $p(y, \mathbf{x}, \tilde{\mathbf{x}}) = p(\mathbf{x}, \tilde{\mathbf{x}}) p(y \mid \mathbf{x})$

$$
\begin{aligned}
- I(\tilde{\mathbf{x}}; y) &= -\mathbb{E}_{p(\mathbf{x}, \tilde{\mathbf{x}}, y)} \Big[\log \frac{p(y \mid \tilde{\mathbf{x}})}{p(y)} \Big] && (1) \\
&= \mathbb{E}_{p(\mathbf{x}, \tilde{\mathbf{x}})} \Big[-\sum_y p(y \mid \mathbf{x}) \log \frac{p(y \mid \tilde{\mathbf{x}})}{p(y)} + \sum_y p(y \mid \mathbf{x}) \log \frac{p(y \mid \mathbf{x})}{p(y \mid \mathbf{x})} \Big] && (2) \\
&= \mathbb{E}_{p(\mathbf{x}, \tilde{\mathbf{x}})} \Big[\sum_y p(y \mid \mathbf{x}) \log \frac{p(y \mid \mathbf{x})}{p(y \mid \tilde{\mathbf{x}})} - \sum_y p(y \mid \mathbf{x}) \log \frac{p(y \mid \mathbf{x})}{p(y)} \Big] && (3) \\
&= \mathbb{E}_{p(\mathbf{x}, \tilde{\mathbf{x}})} \Big[\mathcal{D}_{\mathrm{KL}} \big(p(y \mid \mathbf{x}) \| p(y \mid \tilde{\mathbf{x}}) \big) - \mathcal{D}_{\mathrm{KL}} \big(p(y \mid \mathbf{x}) \| p(y) \big) \Big] && (4)
\end{aligned}
$$

Self-supervised DA with Consistency Training

The first item is known as the Kullback-Leibler consistency loss as considered by VAT [Gidaris, 2018] and UDA [Xie, 2019] for semi-supervised learning:

$$\mathcal{L}_c(\boldsymbol{\theta_e}, \boldsymbol{\theta_m}) = \mathbb{E}_{\mathbf{x}^t \in D_t} \mathbb{E}_{\tilde{\mathbf{x}}^t \in \tilde{D}_t} \left[\mathcal{D}_{\mathrm{KL}}(\hat{p}(y^t \mid \mathbf{x}^t) \| p(y^t \mid \tilde{\mathbf{x}^t})) \right]$$

Self-supervised DA with Consistency Training

The first item is known as the Kullback-Leibler consistency loss as considered by VAT [Gidaris, 2018] and UDA [Xie, 2019] for semi-supervised learning:

$$\mathcal{L}_c(\boldsymbol{\theta_e}, \boldsymbol{\theta_m}) = \mathbb{E}_{\mathbf{x}^t \in D_t} \mathbb{E}_{\tilde{\mathbf{x}}^t \in \tilde{D}_t} \left[\mathcal{D}_{\mathrm{KL}}(\hat{p}(y^t \mid \mathbf{x}^t) \| p(y^t \mid \tilde{\mathbf{x}^t})) \right]$$

Assuming that p(y) is uniform, then the second term can be simplified as:

$$\mathbb{E}_{\mathbf{x}}[-\sum_y p(y \mid \mathbf{x}) \log p(y \mid \mathbf{x})],$$

Self-supervised DA with Consistency Training

The first item is known as the Kullback-Leibler consistency loss as considered by VAT [Gidaris, 2018] and UDA [Xie, 2019] for semi-supervised learning:

$$\mathcal{L}_c(\boldsymbol{\theta_e}, \boldsymbol{\theta_m}) = \mathbb{E}_{\mathbf{x}^t \in D_t} \mathbb{E}_{\tilde{\mathbf{x}}^t \in \tilde{D}_t} \Big[\mathcal{D}_{\mathrm{KL}}(\hat{p}(y^t \mid \mathbf{x}^t) \| p(y^t \mid \tilde{\mathbf{x}^t})) \Big]$$

Assuming that p(y) is uniform, then the second term can be simplified as:

$$\mathbb{E}_{\mathbf{x}}[-\sum_y p(y \mid \mathbf{x}) \log p(y \mid \mathbf{x})],$$

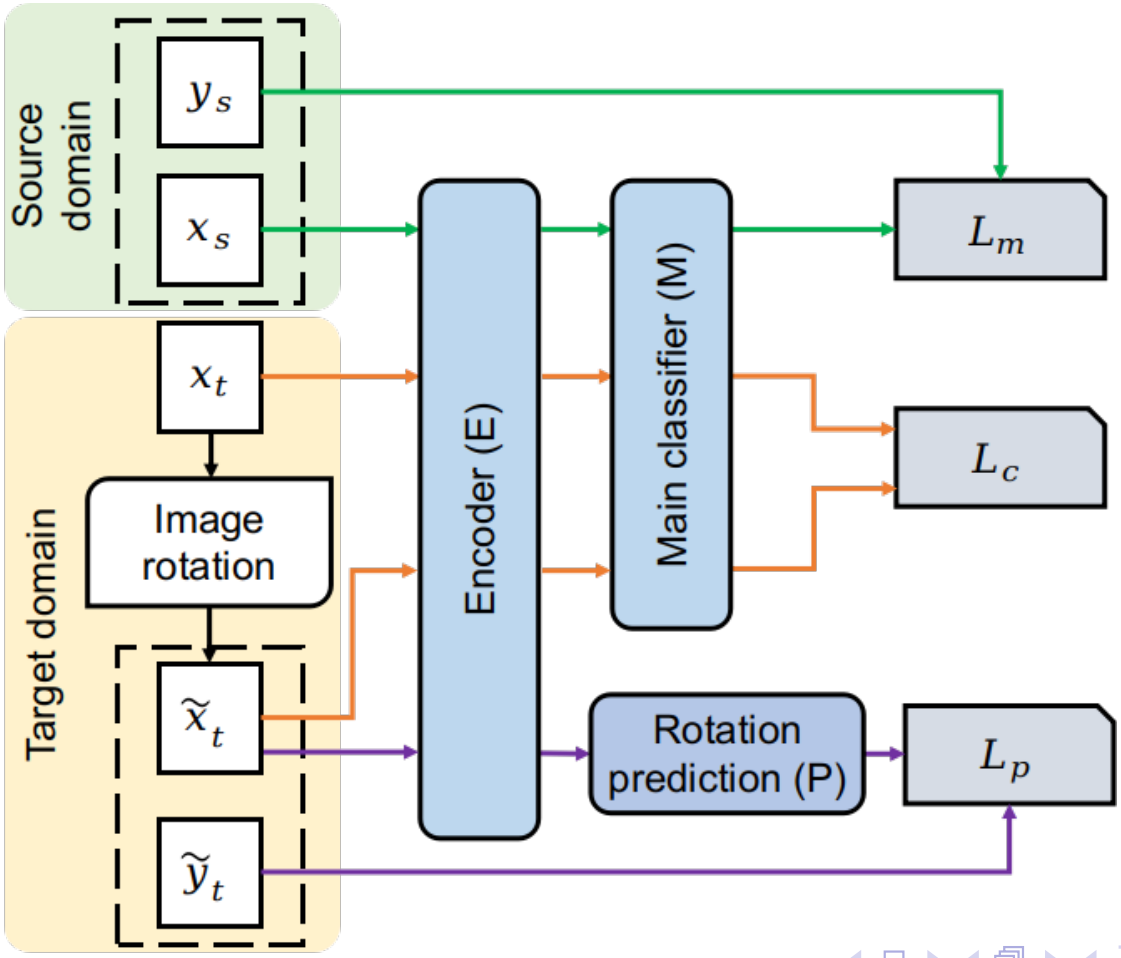
which suggests the following entropy minimization loss term:

$$\mathcal{L}_e(\boldsymbol{\theta_e}, \boldsymbol{\theta_m}) = \mathbb{E}_{\mathbf{x}^t \in D_t} \Big[-\sum_{y^t} p(y^t \mid \mathbf{x}^t) \log p(y^t \mid \mathbf{x}^t) \Big],$$

Self-supervised DA with Consistency Training

Final objective function:

$$\min_{\boldsymbol{\theta_e},\boldsymbol{\theta_m},\boldsymbol{\theta_p}} \mathcal{L}_m(\boldsymbol{\theta_e},\boldsymbol{\theta_m}) + \lambda_p \mathcal{L}_p(\boldsymbol{\theta_e},\boldsymbol{\theta_p}) + \lambda_c \mathcal{L}_c(\boldsymbol{\theta_e},\boldsymbol{\theta_m}) + \lambda_e \mathcal{L}_e(\boldsymbol{\theta_e},\boldsymbol{\theta_m})$$



Experiments

Table: Multi-source Domain Adaptation results on PACS (ResNet-18). Three domains are used as source datasets and the remaining one as target.

Method	Art.	Cartoon	Sketch	Photo	Avg.
SRC	74.7	72.4	60.1	92.9	75.0
Dial	87.3	85.5	66.8	97.0	84.2
DDiscovery	87.7	86.9	69.6	97.0	85.3
CDAN	85.7	88.1	73.1	97.2	86.0
CDAN+E	87.4	**89.4**	**75.3**	97.8	87.5
JiGen	84.9	81.1	79.1	97.9	85.7
Jigsaw	84.9	83.9	69.0	93.9	82.9
Rot	88.7	86.4	74.9	**98.0**	87.0
Ours	**90.3**	87.4	75.1	97.9	**87.7**

Experiments

Table: Accuracy (%) on Office-31 dataset (ResNet-50).

Method	$A \to W$	$D \to W$	$W \to D$	$A \to D$	$D \to A$	$W \to A$	Avg.
ResNet-50	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
DANN	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
CDAN	93.1±0.2	98.2±0.2	**100.0**±0.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
CDAN+E	**94.1**±0.1	**98.6**±0.1	**100.0**±0.0	**92.9**±0.2	71.0±0.3	69.3±0.3	**87.7**
Jigsaw	86.9±0.8	**98.6**±0.5	**100.0**±0.0	82.9±1.0	62.9±1.2	61.2±0.7	82.1
Rot	90.1±0.8	98.1±0.3	**100.0**±0.0	88.6±0.7	65.1±0.8	65.0±0.6	84.5
Ours	92.5±0.2	**98.7**±0.3	**100.0**±0.0	88.6±0.2	69.4±0.4	67.2±0.3	86.1

Experiments

Table: Accuracy (%) on Image-CLEF (ResNet-50).

Method	$I \to P$	$P \to I$	$I \to C$	$C \to I$	$C \to P$	$P \to C$	Avg.
ResNet-50	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN	74.5±0.4	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
DANN	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
JAN	76.8±0.4	88.0±0.2	94.7±0.2	89.5±0.3	74.2±0.3	91.7±0.3	85.8
CDAN	76.7±0.3	90.6±0.3	97.0±0.4	90.5±0.4	**74.5**±0.3	93.5±0.4	87.1
CDAN+E	77.7±0.3	90.7±0.2	**97.7**±0.3	**91.3**±0.3	74.2±0.2	94.3±0.3	**87.7**
Rot	77.9±0.8	91.6±0.3	95.6±0.2	86.9±0.6	70.5±0.7	94.8±0.3	84.2
Ours	**78.6**±0.4	**92.5**±0.1	96.1±0.3	88.9±0.2	73.9±0.7	**95.9**±0.6	**87.7**

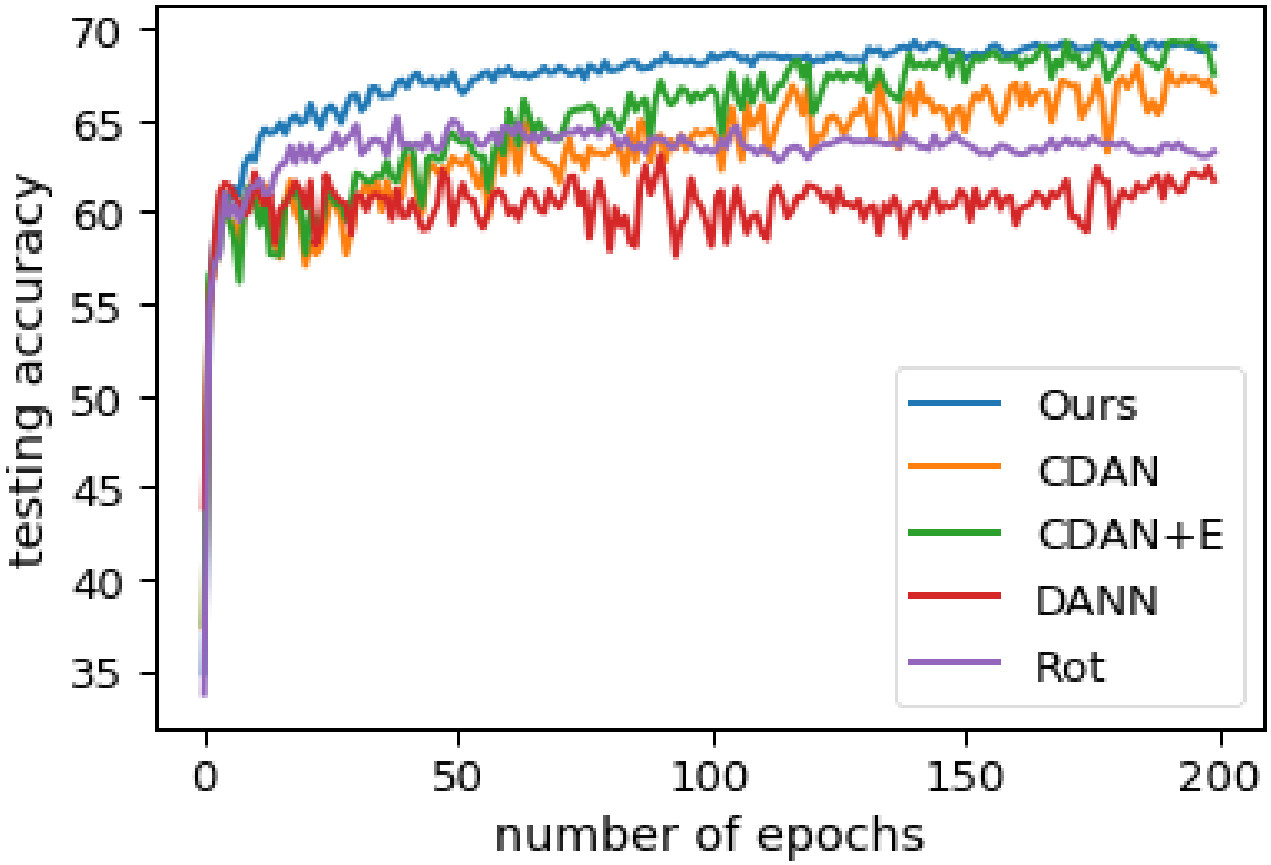
Experiments

Table: Accuracy (%) on Office-Home (ResNet-50).

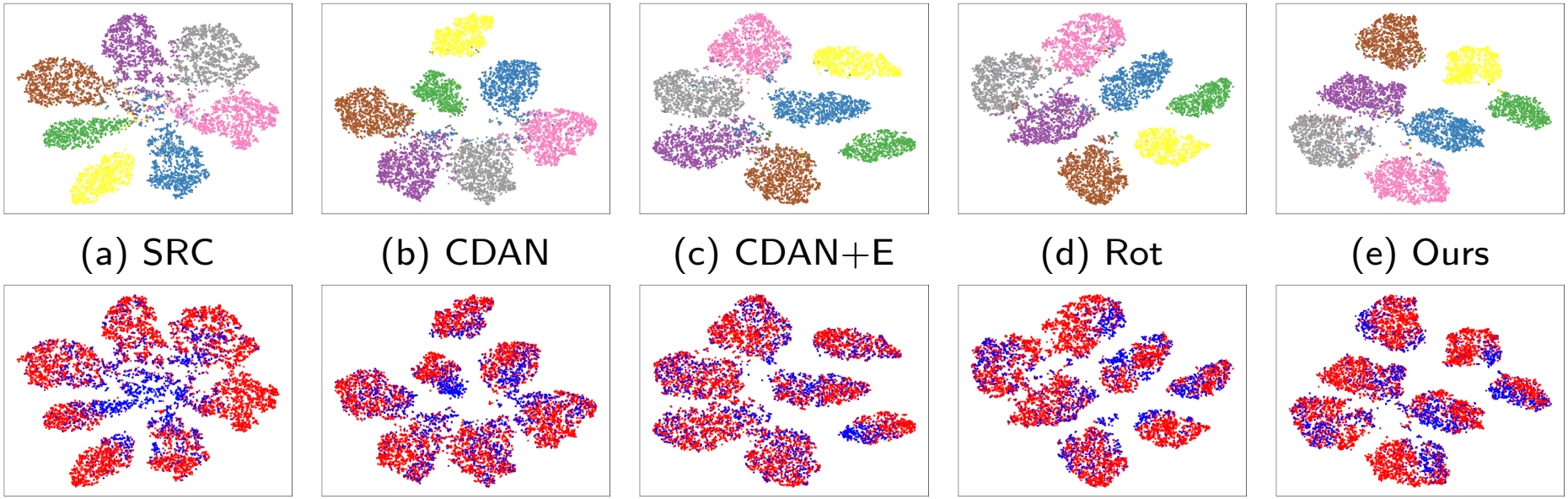
Method	*ar*→*cl*	*ar*→*pr*	*ar*→*rw*	*cl*→*ar*	*cl*→*pr*	*cl*→*rw*	*pr*→*ar*	*pr*→*cl*	*pr*→*rw*	*rw*→*ar*	*rw*→*cl*	*rw*→*pr*	Avg.
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	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	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E	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
Rot	50.4	67.8	74.6	58.7	66.7	67.4	55.7	52.4	77.5	71.0	59.6	81.2	65.3
Ours	**51.7**	69.0	75.4	**60.4**	**70.3**	**70.7**	**57.7**	**53.3**	**78.6**	**72.2**	**59.9**	**81.7**	**66.7**

Convergence analysis

Table: Convergence of different methods on task $D \rightarrow A$.



Feature Visualization



(a) SRC (b) CDAN (c) CDAN+E (d) Rot (e) Ours

(f) SRC (g) CDAN (h) CDAN+E (i) Rot (j) Ours

Figure: The t-SNE visualization of deep features in PACS DA task (art painting is used as target domain). (a)-(e) are feature distribution visualization with category colors. (f)-(j) are feature distribution visualization with domain colors. Red and blue points represent samples of source and target domains, respectively.

References

- J. Xu, L. Xiao, and A. M. Lopez. Self-supervised domain adaptation for computer vision tasks. *IEEE Access.* vol. 7, 156694 – 156706, 2019.
- Q. Xie, Z. Dai, E. Hovy, M. Luong, and Q. V. Le. Unsupervised data augmentation for consistency training. *arXiv.* 2019.
- S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR.* 2018.
- T. Miyato, S. ichi Maeda, S. Ishii, and M. Koyama, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, *IEEE T-PAMI*, 2018

Thank you

Code available at: https://github.com/Jiaolong/ss-da-consistency.