CaPAN: Class-aware Prototypical Adversarial Networks for Unsupervised Domain Adaptation

Zhenyu Yu[1,4], Pei Wang[2,3*],

[1]Guangzhou University, China

[2]National Pilot School of Software, Yunnan University, China

[3]School of Information Science and Engineering, Yunnan University, China

[4] Universiti Malaya, Malaysia

Contributions

- **Class-aware adversarial learning**: single multi-class classifier is served as a discriminator to align class-level feature distributions

- **Prototypical-based domain discriminator**: generate more compact features in an adversarial paradigm

- **Gradient analysis** for different adversarial learning paradigm

Code@: https://github.com/YuZhenyuLindy/CaPAN

Class-aware Prototypical Adversarial Networks

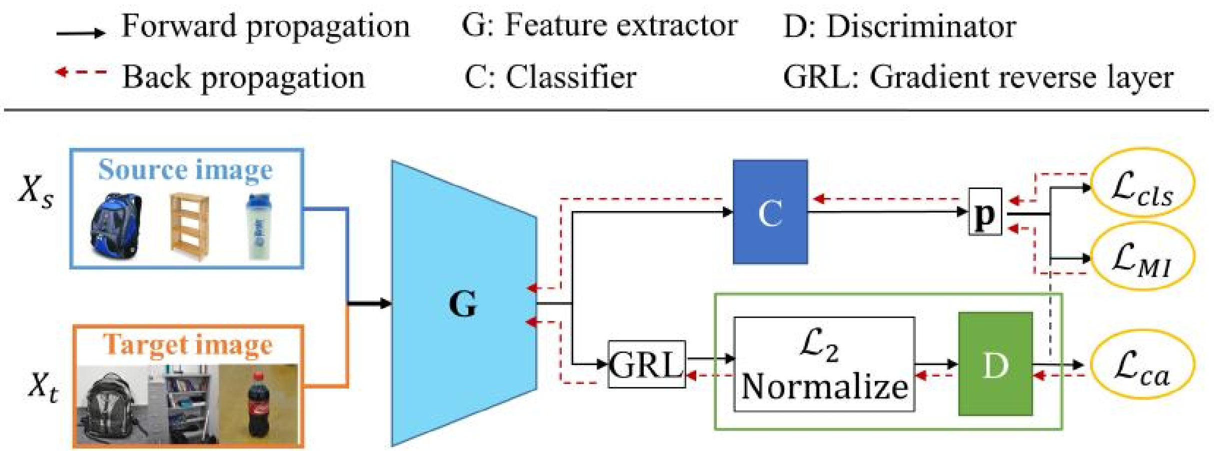


Fig. 1: The architecture of the proposed Class-aware Prototypical Adversarial Networks (CaPAN), where $\mathbf{p}$ is the predicted output. The connection with $\mathbf{p}$ and $\mathcal{L}_{ca}$ is represented by a dashed line with a weighted meaning.

Class-aware Prototypical Adversarial Networks

Class-Aware Adversarial Learning

$$\min_{G}\max_{D}\mathcal{L}_{ca}=\frac{1}{n_s}\sum_{i=1}^{n_s}\log\left[D_{\boldsymbol{y}_i^s}\left(G\left(\boldsymbol{x}_i^s\right)\right)\right]$$

$$+\frac{1}{n_t}\sum_{j=1}^{n_t}\sum_{k=1}^{K}\boldsymbol{p}_k^t(\boldsymbol{x}_j^t)\log\left[1-D_k\left(G\left(\boldsymbol{x}_i^t\right)\right)\right]$$

Prototypical domain discriminator

weight $\mathbf{W}=[w_1, w_2, \cdots, w_K]$

probability $p(\boldsymbol{x})=\sigma(\frac{\mathbf{W}^{\mathrm{T}}\boldsymbol{z}}{T\|\boldsymbol{z}\|})\in\mathbb{R}^K$

Class-aware Prototypical Adversarial Networks

Overall Formulation

$$\min_{G,C}\max_{D} \mathcal{L}_{CaPAN} = \mathcal{L}_{cls} + \beta\mathcal{L}_{ca} - \lambda\mathcal{L}_{MI}$$

$$\max_{G,C} \mathcal{L}_{MI} = H(\boldsymbol{y}) - H(\boldsymbol{y} \mid \boldsymbol{x})$$

$$= -\sum_{k=1}^{K} \overline{\boldsymbol{p}}_k^t \log \overline{\boldsymbol{p}}_k^t + \frac{1}{N_t}\sum_{j=1}^{N_t}\sum_{k=1}^{K} \boldsymbol{p}_{jk}^t \log \boldsymbol{p}_{jk}^t,$$

Regularizer Loss

$$\mathcal{L}_{reg} = \mathcal{L}_{CaPAN} + \gamma\mathcal{L}_{spe}$$

$\mathcal{L}_{spe}$: specific losses used in existing methods

Gradient analysis

Gradient for DANN

$$\frac{\partial \log[1-D(\boldsymbol{z})]}{\partial \boldsymbol{z}} = \frac{-1}{1-D(\boldsymbol{z})} D(\boldsymbol{z})\left(1-D(\boldsymbol{z})\right)\frac{\partial \phi_D(\boldsymbol{z})}{\partial \boldsymbol{z}}$$

$$= -D(\boldsymbol{z})\frac{\partial \phi_D(\boldsymbol{z})}{\partial \boldsymbol{z}} = -D(\boldsymbol{z})\boldsymbol{w}_D$$

Gradient for Class-Aware adversarial Learning

$$\frac{\partial \log[1-D_k(\boldsymbol{z})]}{\partial \boldsymbol{z}} = \frac{-1}{1-D_k(\boldsymbol{z})}\frac{\partial D_k(\boldsymbol{z})}{\partial \boldsymbol{z}}$$

$$= \frac{-D_k(\boldsymbol{z})}{1-D_k(\boldsymbol{z})}\left(\boldsymbol{w}_k - \sum_{k'=1}^{K} \boldsymbol{w}_{k'} D_{k'}(\boldsymbol{z})\right)$$

Theoretical Insight

generalization error upper bound on target domain

$$\epsilon_t(h) \le \epsilon_s(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(P_s, P_t) + \epsilon_{ideal}(h^*)$$

$\epsilon_s(h)$: expected error on source domain

$d_{\mathcal{H}\Delta\mathcal{H}}(P_s, P_t)$: distribution discrepancy

$\epsilon_{ideal}(h^*)$: joint error of the ideal hypothesis

Results

TABLE I: Accuracy (%) on (a) Office-31 (ResNet-50) and (b) VisDA-2017 (ResNet-101) for UDA. † denotes that the results are cited from [26]. * denotes that the results are reproduced using the publicly released code.

(a) Office-31.

Method	A→W	D→W	W→D	A→D	D→A	W→A	AVG
ResNet-50 [1]	68.4±0.5	96.7±0.5	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN [3]	83.8±0.4	96.8±0.2	99.5±0.1	78.4±0.2	66.7±0.3	62.7±0.2	81.3
DANN [7]	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
DWL [20]	89.2	99.2	**100.0**	91.2	73.1	69.8	87.1
MADA [13]	90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
DPN [27]	91.5±0.4	**99.5±0.5**	**100.0±0.0**	94.0±0.9	72.2±1.3	68.1±0.1	87.6
GATE [21]	90.5	98.7	**100.0**	91.3	73.4	75.9	88.3
DSAN [16]	93.6±0.2	98.3±0.1	**100.0±0.0**	90.2±0.7	73.5±0.5	74.8±0.4	88.4
MDD [18]	94.5±0.3	98.4±0.1	**100.0±0.0**	93.5±0.2	74.6±0.3	72.2±0.1	88.9
MCC [22]	95.5	98.6	**100.0**	94.4	72.9	74.9	89.4
SCDA [24]	94.2	98.7	99.8	95.2	75.7	**76.2**	90.0
CaPAN(w/ classifier)	**95.7**	98.7	99.9	94.0	76.1	75.3	90.0
CaPAN(w/ sigmoid)	94.8	98.8	99.9	94.2	77.0	75.8	90.1
CaPAN(Ours)	94.8±0.5	98.7±0.3	**100.0±0.0**	**95.4±0.6**	**77.1±0.7**	76.1±0.1	**90.3**

(b) VisDA-2017.

Method	AVG
ResNet-101 [1]	52.4
DAN [3]	61.1
DANN [7]	57.4
DWL [20]	77.1
MCD [9]	71.9
CDAN [8]	73.9
GATE [21]	74.8
DSAN [16]	75.1
SWD [10]	76.4
TSA [23]	78.6
SCDA† [26]	79.7
DADA [15]	79.8
CaPAN(Ours)	80.5
MDD*	77.3
MDD+CaPAN	81.3(4.0↑)
MCC [22]	78.8
MCC+CaPAN	**83.5(4.7↑)**

TABLE II: Accuracy (%) on Office-Home for UDA (ResNet-50).

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	AVG
ResNet-50 [1]	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
DANN [7]	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 [28]	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
DSAN [16]	54.4	70.8	75.4	60.4	67.8	68.0	62.6	55.9	78.5	73.8	**60.6**	83.1	67.6
MDD [18]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
MCC [22]	55.1	75.2	79.5	63.3	73.2	**75.8**	**66.1**	52.1	76.9	73.8	58.4	83.6	69.4
DPN [27]	51.8	75.3	79.4	66.6	74.8	74.6	63.8	51.7	**81.5**	74.0	58.0	84.3	69.7
GATE [21]	54.6	76.9	79.8	66.1	73.5	74.2	65.3	54.8	80.6	73.9	59.5	83.7	70.2
SCDA [24]	**57.5**	76.9	80.3	65.7	74.9	74.5	65.5	53.6	79.8	**74.5**	59.6	83.7	70.5
CaPAN(Ours)	56.6	**77.0**	**81.2**	**66.7**	**76.8**	**75.8**	65.2	**56.5**	81.4	74.3	59.3	**84.5**	**71.3

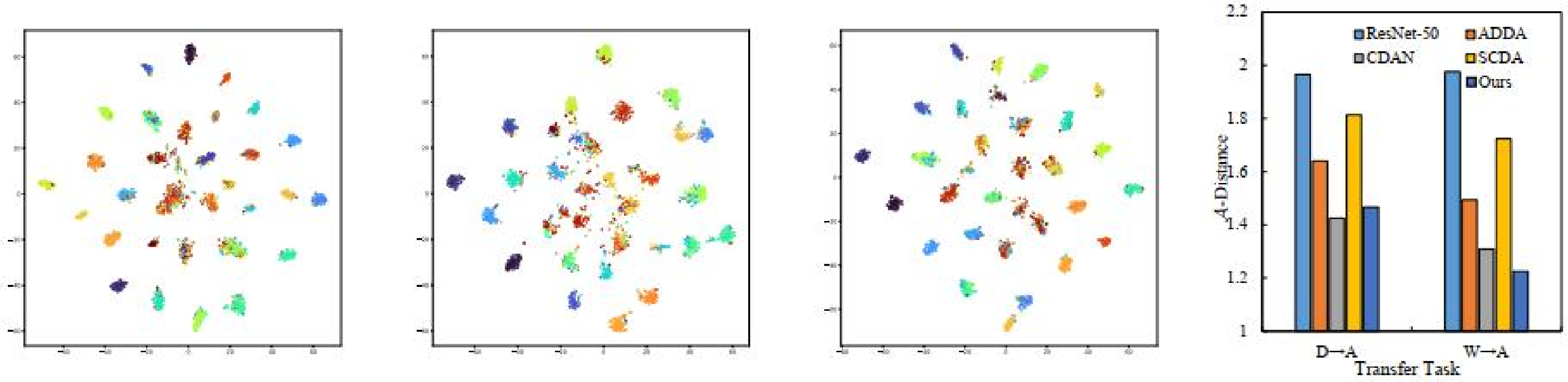
Analysis

Ablation study

TABLE I: Accuracy (%) on (a) Office-31 (ResNet-50)

Method	A→W	D→W	W→D	A→D	D→A	W→A	AVG
CaPAN(w/ classifier)	**95.7**	98.7	99.9	94.0	76.1	75.3	90.0
CaPAN(w/ sigmoid)	94.8	98.8	99.9	94.2	77.0	75.8	90.1
CaPAN	94.8±0.5	98.7±0.3	**100.0±0.0**	**95.4±0.6**	**77.1±0.7**	76.1±0.1	**90.3**

Feature Visualization & Distribution Discrepancy



(a) CDAN (b) SCDA (c) CaPAN (d) $\mathcal{A}$-Distance

Fig. 2: (a)-(c) Feature visualization of different methods of the target domain on task D→A. (d) Distribution analysis of task D→A and W→A.

Thanks!

CaPAN: Class-aware Prototypical Adversarial Networks for Unsupervised Domain Adaptation

Zhenyu Yu, Pei Wang*

Code

Email