Partial Adversarial Domain Adaptation

Zhangjie Cao, Lijia Ma, Mingsheng Long and Jianmin Wang
KLiss, MOE; NEL-BDS; TNList; School of Software, Tsinghua University, China

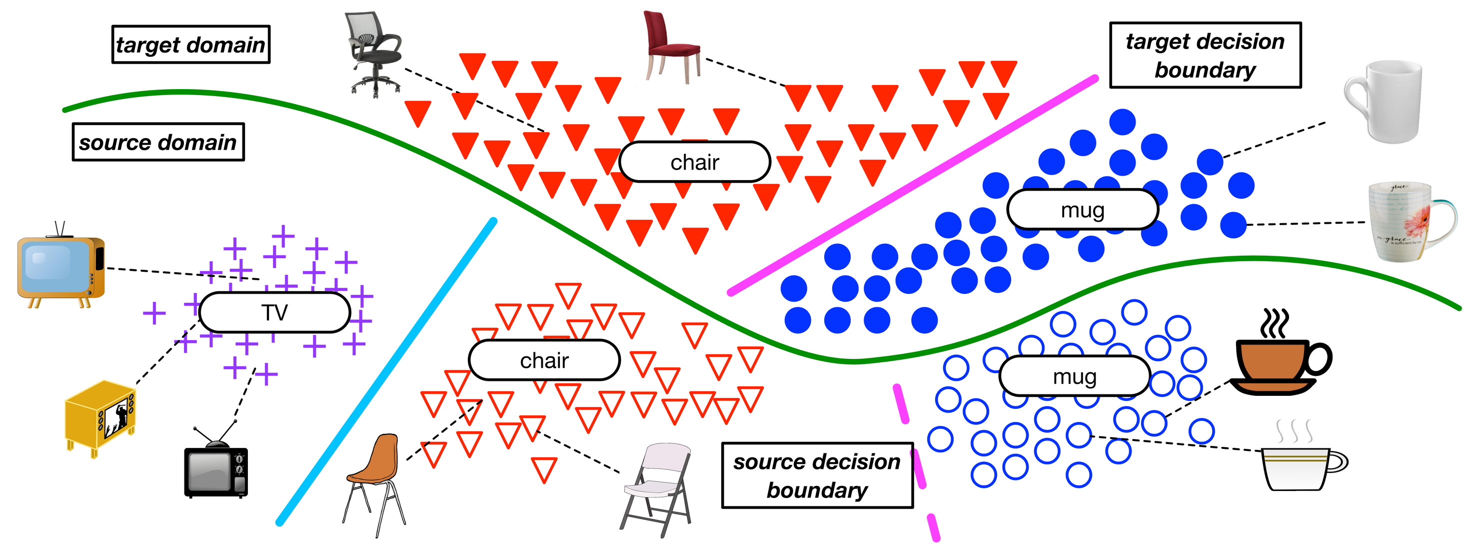
Summary

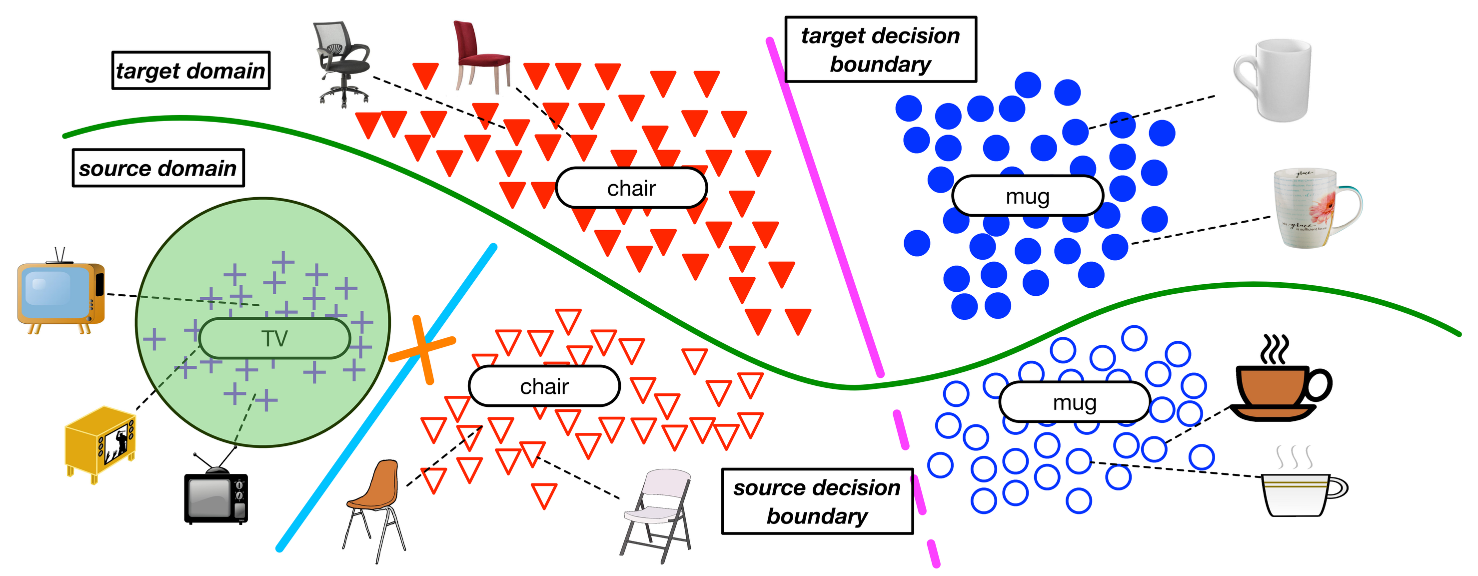
- Partial domain adaptation: Deep learning across domains with different label spaces $\mathcal{C}_s \supset \mathcal{C}_t$
- Two main challenges:
  - Positive transfer across domains in shared label space $P_{\mathcal{C}_t} \neq Q_{\mathcal{C}_t}$
  - Negative transfer across domains in outlier label space $P_{\mathcal{C}_s \backslash \mathcal{C}_t} \neq Q_{\mathcal{C}_t}$
- State-of-the-art results on partial domain adaptation datasets.
- Main contributions:
  - Propose a class-wise weight driven from the prediction of target data by the source classifier;
  - Apply the proposed weighting mechanism to both the source classifier and the domain adversarial network to avoid negative transfer.
- Code available @ `https://github.com/thuml/PADA`

Partial Domain Adaptation

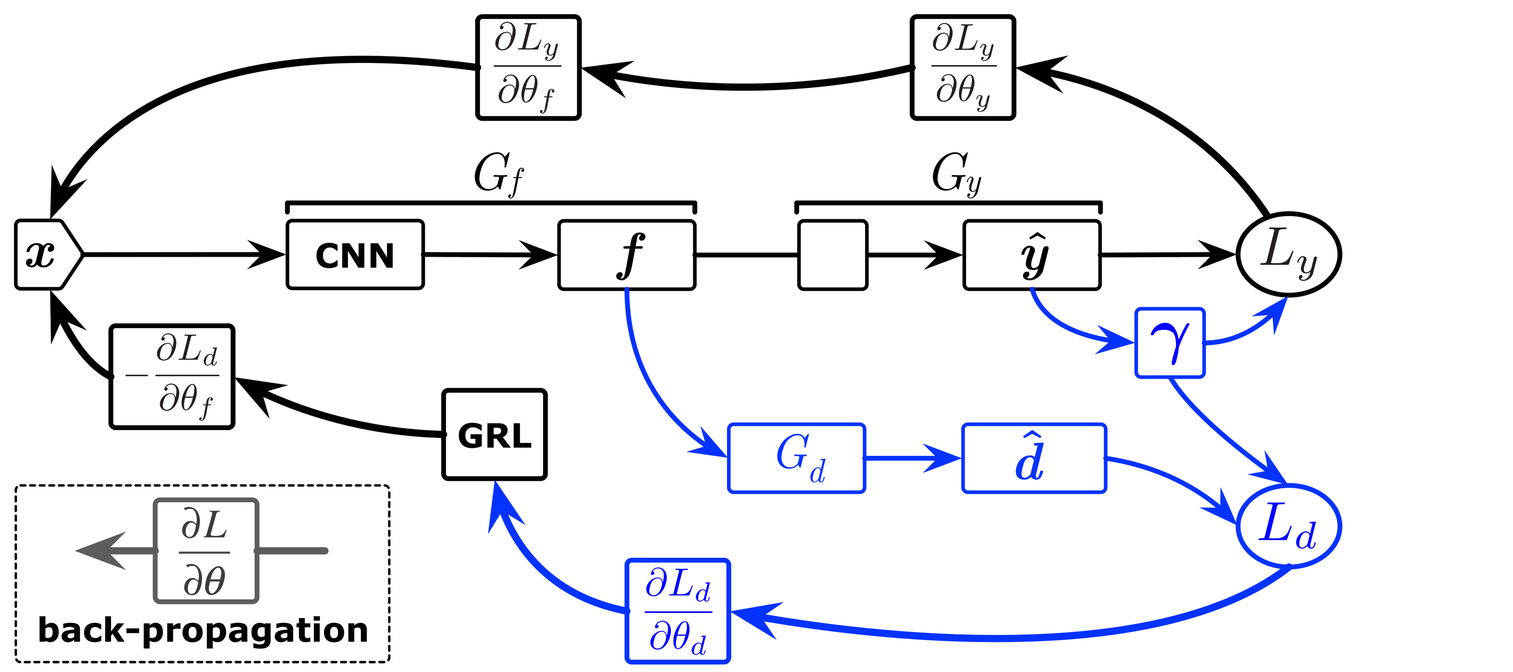
- Deep learning across domains with different label spaces $\mathcal{C}_s \supset \mathcal{C}_t$
- Positive transfer across domains in shared label space $P_{\mathcal{C}_t} \neq Q_{\mathcal{C}_t}$
- Negative transfer across domains in outlier label space $P_{\mathcal{C}_s \backslash \mathcal{C}_t} \neq Q_{\mathcal{C}_t}$

Partial Domain Adaptation: How?





Partial Adversarial Domain Adaptation



- $\mathbf{f} = G_f(\mathbf{x})$: feature extractor
- γ : class weight averaged on the label predictions of target data
- GRL: gradient reversal layer
- G_y, L_y, $\hat{\mathbf{y}}$: label predictor, loss and predicted class label
- G_d, L_d, $\hat{\mathbf{d}}$: domain discriminator, loss and predicted domain label

Weighting Mechanism and Loss

- Weighting mechanism

$$\gamma = \frac{1}{n_t}\sum_{i=1}^{n_t} \hat{\mathbf{y}}_i, \tag{1}$$

where n_t is #target data and $\hat{\mathbf{y}}_i$ is the label prediction of target data i.

- Overall loss

$$\begin{aligned} C\left(\theta_f, \theta_y, \theta_d\right) = & \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \gamma_{y_i} L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right), y_i\right) \\ & - \frac{\lambda}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \gamma_{y_i} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right) \\ & - \frac{\lambda}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right) \end{aligned} \tag{2}$$

where n_s is #source data, $\mathcal{C}_s$ is the source domain and $\mathcal{C}_t$ is the target domain.

- Optimization in a domain adversarial learning framework

$$\begin{aligned} \left(\hat{\theta}_f, \hat{\theta}_y\right) &= \arg\min_{\theta_f, \theta_y} C\left(\theta_f, \theta_y, \theta_d\right), \\ \left(\hat{\theta}_d\right) &= \arg\max_{\theta_d} C\left(\theta_f, \theta_y, \theta_d\right). \end{aligned} \tag{3}$$

where θ_f, θ_y and θ_d are parameters of G_f, G_y and G_d. $\hat{\theta}_f$, $\hat{\theta}_y$ and $\hat{\theta}_d$ are the corresponding optimal parameters.

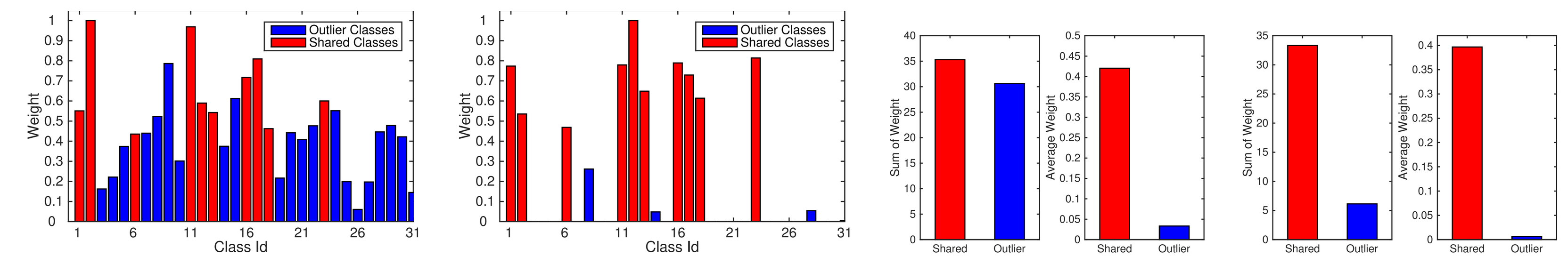
Experimental Results

Table: Accuracy of partial domain adaptation tasks on **Office-Home**, **Office-31**, **ImageNet-Caltech** and **VisDA2017** (ResNet-50)

Method	Office-Home												
	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	38.57	60.78	75.21	39.94	48.12	52.90	49.68	30.91	70.79	65.38	41.79	70.42	53.71
DAN	44.36	61.79	74.49	41.78	45.21	54.11	46.92	38.14	68.42	64.37	45.37	68.85	54.48
DANN	44.89	54.06	68.97	36.27	34.34	45.22	44.08	38.03	68.69	52.98	34.68	46.50	47.39
RTN	49.37	64.33	76.19	47.56	51.74	57.67	50.38	41.45	75.53	70.17	51.82	74.78	59.25
PADA	**51.95**	**67**	**78.74**	**52.16**	**53.78**	**59.03**	**52.61**	**43.22**	**78.79**	**73.73**	**56.6**	**77.09**	**62.06**

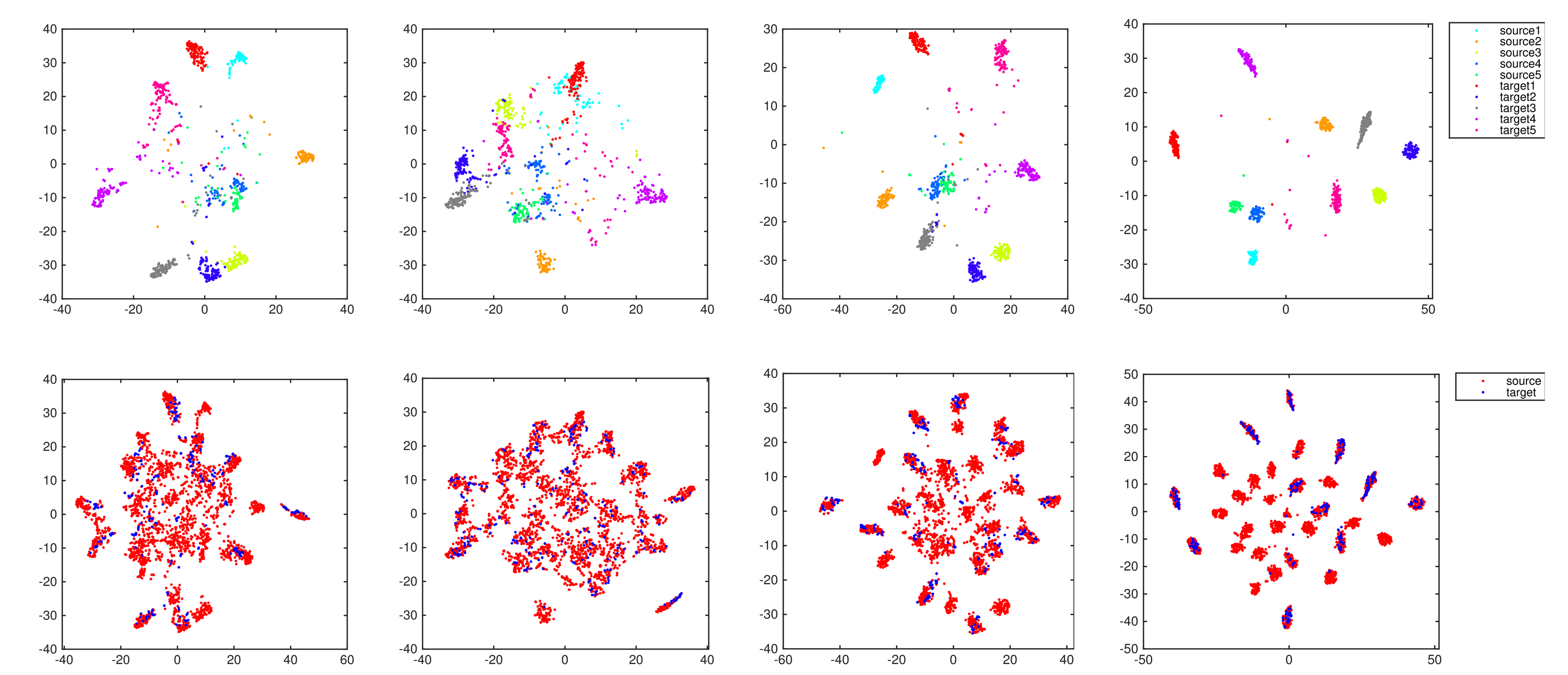
Method	Office-31						
	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet	54.52	94.57	94.27	65.61	73.17	71.71	75.64
DAN	46.44	53.56	58.60	42.68	65.66	65.34	55.38
DANN	41.35	46.78	38.85	41.36	41.34	44.68	42.39
ADDA	43.65	46.48	40.12	43.66	42.76	45.95	43.77
RTN	75.25	97.12	98.32	66.88	85.59	85.70	84.81
JAN	43.39	53.56	41.40	35.67	51.04	51.57	46.11
LEL	73.22	93.90	96.82	76.43	83.62	84.76	84.79
PADA-classifier	83.12	99.32	100	80.16	90.13	92.34	90.85
PADA-adversarial	65.76	97.29	97.45	77.07	87.27	87.37	85.37
PADA	**86.54**	**99.32**	**100**	**82.17**	**92.69**	**95.41**	**92.69**

Method	ImageNet-Caltech			VisDA2017		
	ImageNet → Caltech	Caltech → ImageNet	Avg	Real → Synthetic	Synthetic → Real	Avg
ResNet	71.65	66.14	68.90	64.28	45.26	54.77
DAN	71.57	66.48	69.03	68.35	47.60	57.98
DANN	68.67	52.97	60.82	73.84	51.01	62.43
RTN	72.24	68.33	70.29	72.93	50.04	61.49
PADA	**75.03**	**70.48**	**72.76**	76.50	53.53	65.01



(a) DANN: Office (b) PADA: Office (c) DANN: ImageNet (d) PADA: ImageNet

Figure: Histograms of class weights learned by PADA and DANN.



(e) DAN (f) RevGrad (g) RTN (h) SAN

Figure: The t-SNE visualization of DAN, RevGrad, RTN, and SAN.