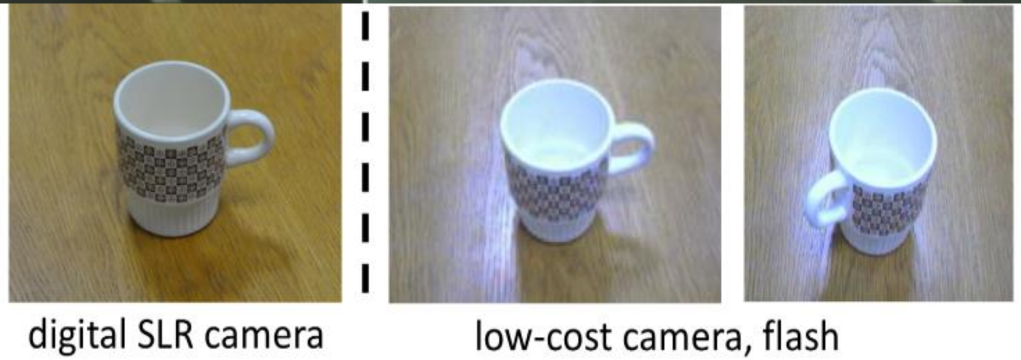
Partial Adversarial Domain Adaptation

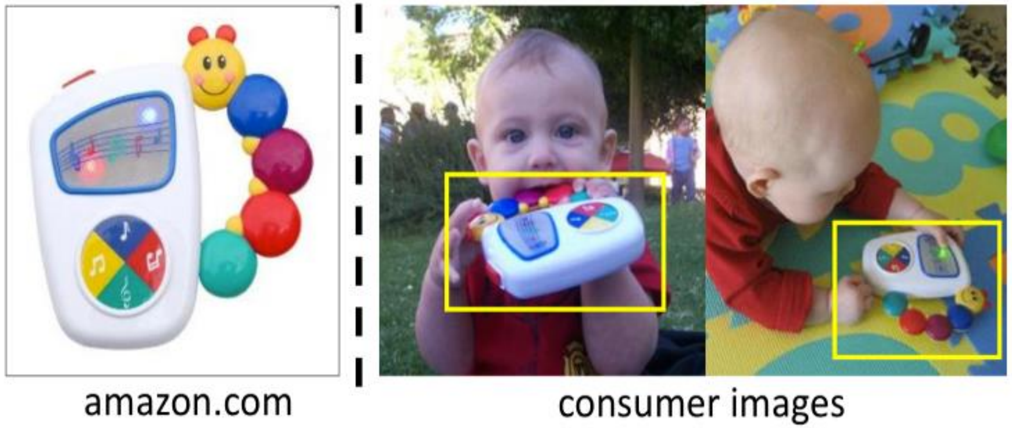
2019-09-25

Ying-Peng Tang

目 录

Домаин Адаптатион





Source domain

- A different but relevant domains
- Abundant training examples

Target domain

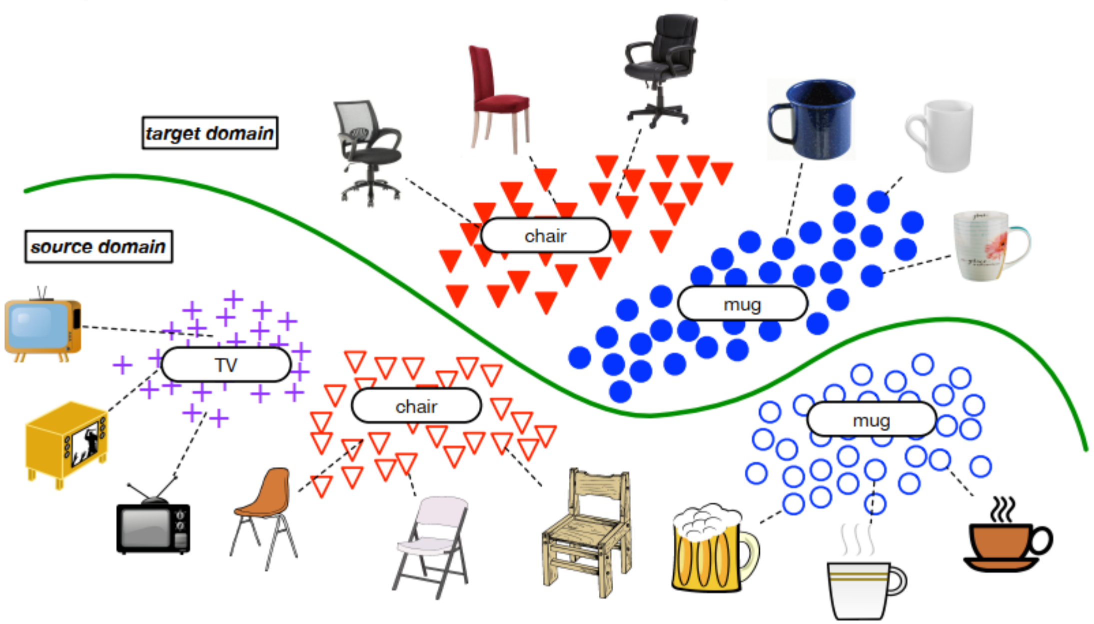
- Target task
- Limited or no training examples

Партиал Домаин Адаптатион

Existing methods generally assume that the source and the target domains share identical label space.

- **Not always satisfied**
- **Cumbersome to seek for source domains for emerging target domains.**

A more practical scenario is that the target label space is a subspace of the source label space.



Адверсариал Домаин Адаптатион

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

L: loss

G_f: Feature extractor

G_y : Classifier

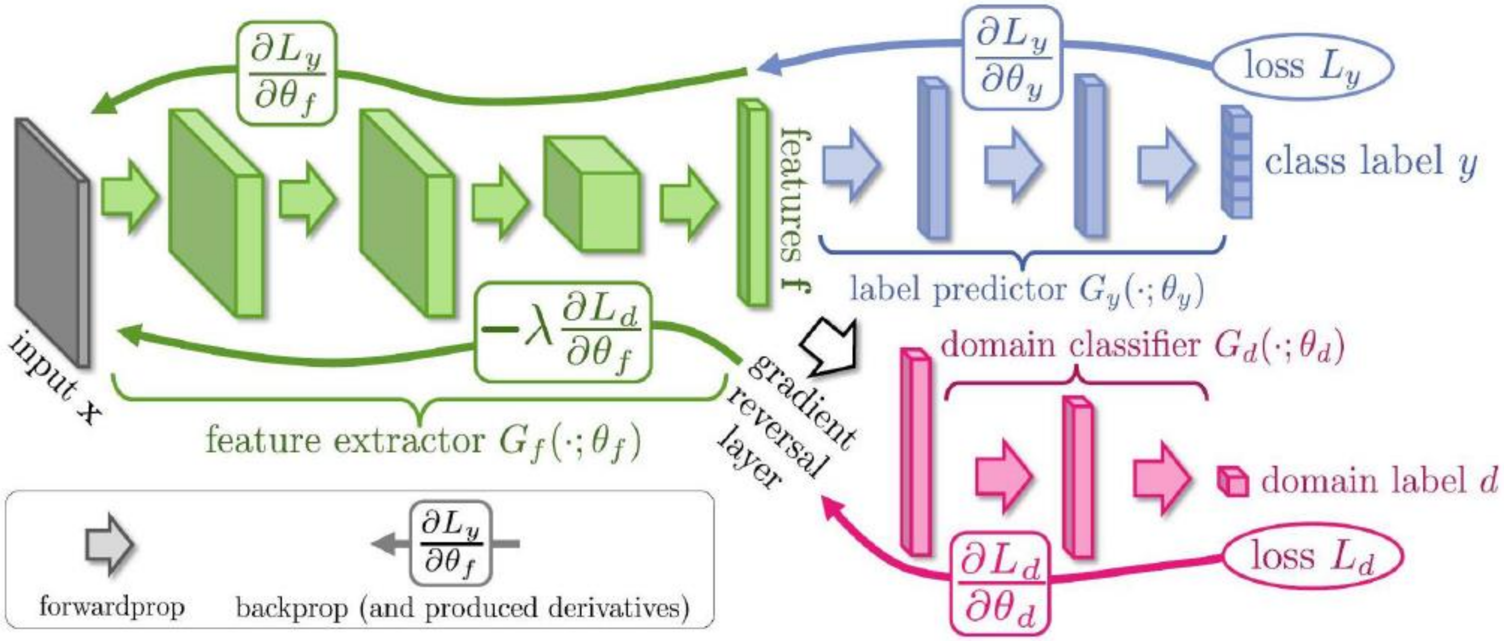
G_d : Domain classifier

x, y: example, label

d: domain label (source 1, target 0)

n: number of examples

Feature extractor



Classifier

Domain classifier

Партиал Домаин Адаптатион

PDA setting : $C_t \subseteq C_s$

Outlier label space : $C_s \backslash C_t$

Target label space : C_t

- Aligning the whole source domain will cause negative transfer since the target domain is also forced to match the outlier label space
- The target domain is unsupervised. We do not know which class is the target class.

Key idea: re-weighting.

ECCV18

Partial Adversarial Domain Adaptation

Zhangjie Cao, Lijia Ma, Mingsheng Long(✉), and Jianmin Wang

School of Software, Tsinghua University, China
National Engineering Laboratory for Big Data Software
Beijing National Research Center for Information Science and Technology

Метход

Key idea: class-level re-weighting. How to find the outlier classes?

Motivation: Since the source outlier label space and target label space are disjoint, the target data should be significantly dissimilar to the source data in the outlier label space

Method: Take the prediction of target data as the class-weight.

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

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

CVPR18

Importance Weighted Adversarial Nets for Partial Domain Adaptation

Jing Zhang, Zewei Ding, Wanqing Li, Philip Ogunbona
Advanced Multimedia Research Lab, University of Wollongong, Australia

Метход

Key idea: image-level re-weighting

Source domain label: 1

Target domain label: 0

For source domain example:

$if\ D^*(z) \approx 1$ highly likely come from the outlier classes

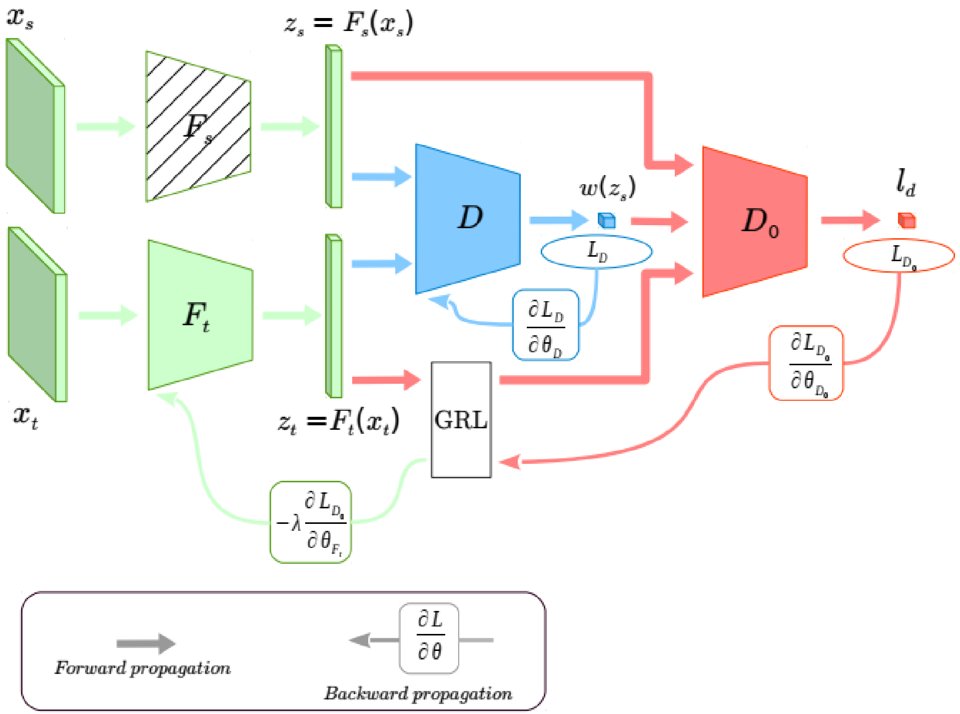
$if\ D^*(z) \approx 0$ more likely come from the shared classes

$$\tilde{w}(\mathbf{z}) = 1 - D^*(\mathbf{z}) = \frac{1}{\frac{p_s(\mathbf{z})}{p_t(\mathbf{z})} + 1}$$

$$w(\mathbf{z}) = \frac{\tilde{w}(\mathbf{z})}{\mathbb{E}_{\mathbf{z} \sim p_s(\mathbf{z})} \tilde{w}(\mathbf{z})}$$

Трицкс

- Adopt the unshared feature extractors for source and target domains
- Initialize Ft using the parameter of Fs
- Use an auxiliary domain classifier D for obtaining the w(z)
- Minimize the entropy of target domain data



$$\min_{F_s,C} \mathcal{L}_s(F_s,C) = -\mathbb{E}_{\mathbf{x},y\sim p_s(\mathbf{x},y)} \sum_{k=1}^{K} \mathbb{1}_{[k=y]} \log C(F_s(\mathbf{x}))$$

$$\min_{D} \mathcal{L}_D(D,F_s,F_t) = -\Big(\mathbb{E}_{\mathbf{x}\sim p_s(\mathbf{x})}[\log D(F_s(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}[\log(1-D(F_t(\mathbf{x})))]\Big)$$

$$\min_{F_t} \max_{D_0} \mathcal{L}_w(C,D_0,F_s,F_t) = \gamma \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})} H(C(F_t(\mathbf{x}))) + \lambda\Big(\mathbb{E}_{\mathbf{x}\sim p_s(\mathbf{x})}[w(\mathbf{z}) \log D_0(F_s(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}[\log(1-D_0(F_t(\mathbf{x})))]\Big)$$

CVPR19

Learning to Transfer Examples for Partial Domain Adaptation

Zhangjie Cao[1]*, Kaichao You[1]*, Mingsheng Long[1](✉), Jianmin Wang[1], and Qiang Yang[2]
[1]KLiss, MOE; BNRist; School of Software, Tsinghua University, China
[1]Research Center for Big Data, Tsinghua University, China
[1]Beijing Key Laboratory for Industrial Big Data System and Application
[2]Hong Kong University of Science and Technology, China

Метход

Key idea: image-level re-weighting.

$$E_{G_y} = \frac{1}{n_s} \sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right) L\left(G_y\left(G_f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right)\right) + \frac{\gamma}{n_t} \sum_{j=1}^{n_t} H\left(G_y\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),$$

$$E_{G_d} = -\frac{1}{n_s} \sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right) \log\left(G_d\left(G_f\left(\mathbf{x}_i^s\right)\right)\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log\left(1 - G_d\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),$$

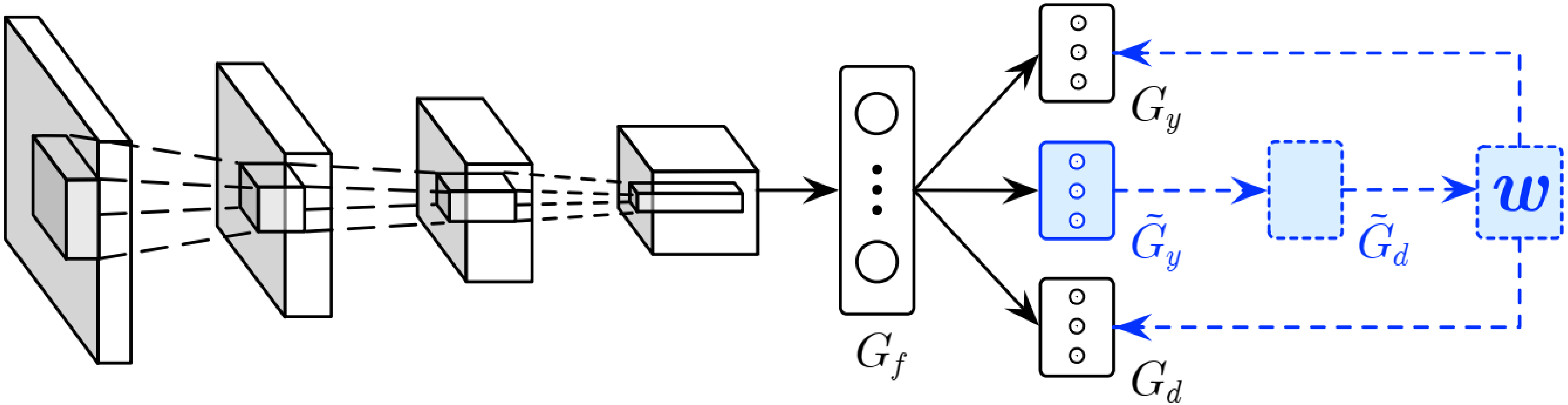
Хощ то гет щ(ш)

- **Consider the domain information (transferability)**

Use an auxiliary domain discriminator $\tilde{G}_d$

- **Consider the discriminative information (relevance)**

Use an auxiliary classifier $\tilde{G}_y$, Within $\tilde{G}_y$, the feature from feature extractor G_f are transformed to $|C_s|$ dimension **z**, Then **z** will be passed through a leaky-softmax activation.



Хощ то гет щ(ш)

$$\tilde{G}_d\left(G_f\left(\mathbf{x}_i\right)\right)=\sum_{c=1}^{|\mathcal{C}_s|}\tilde{G}_y^c\left(G_f\left(\mathbf{x}_i\right)\right) \qquad w\left(\mathbf{x}_i^s\right)=1-\tilde{G}_d\left(G_f\left(\mathbf{x}_i^s\right)\right).$$

$$E_{\tilde{G}_d}=-\frac{1}{n_s}\sum_{i=1}^{n_s}\log\left(\tilde{G}_d\left(G_f\left(\mathbf{x}_i^s\right)\right)\right) \qquad w\left(\mathbf{x}\right)\leftarrow\frac{w(\mathbf{x})}{\frac{1}{B}\sum_{i=1}^{B}w(\mathbf{x}_i)}$$

$$-\frac{1}{n_t}\sum_{j=1}^{n_t}\log\left(1-\tilde{G}_d\left(G_f\left(\mathbf{x}_j^t\right)\right)\right)$$

Motivation: As the auxiliary label predictor $\tilde{G}_y$ is trained on source examples and labels, the prediction for source data should be certain, for target data should be uncertain. Therefore, the element-sum of the leaky-softmax outputs is closer to 1 for source examples and closer to 0 for target examples.

Објецтиве фунцтион

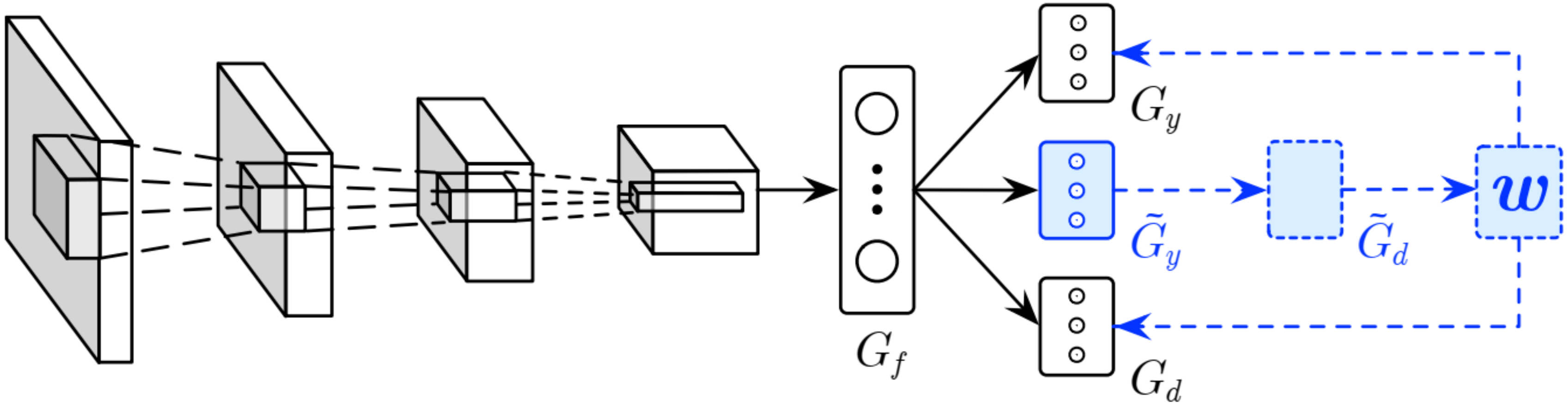
$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E_{G_y} - E_{G_d}$$

$$(\hat{\theta}_d) = \arg\min_{\theta_d} E_{G_d},$$

$$(\hat{\theta}_{\tilde{y}}) = \arg\min_{\theta_{\tilde{y}}} E_{\tilde{G}_y} + E_{\tilde{G}_d}.$$

$$E_{G_y} = \frac{1}{n_s} \sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right) L\left(G_y\left(G_f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right)\right) + \frac{\gamma}{n_t} \sum_{j=1}^{n_t} H\left(G_y\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),$$

$$E_{G_d} = -\frac{1}{n_s} \sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right) \log\left(G_d\left(G_f\left(\mathbf{x}_i^s\right)\right)\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log\left(1 - G_d\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),$$



Dataset

- **Office-31**, 31 categories in total, 3 Domain. 31 categories for the source domain and 10 categories for the target domain. **A→W**, **D → W**, **W → D**, **A → D**, **D → A** and **W → A**
- **Office-Home**, 65 categories, 4 domains: **Ar**, **Cl**, **Pr**, **Rw.** first 25 categories in alphabetical order as the target domain and images from all 65 categories as the source domain
- **ImageNet(1000) → Caltech (84)** They share 84 classes
- **Caltech (256) → ImageNet (84)** They share 84 classes

Compared methods

- ResNet
- Deep Adaptation Network (**DAN**)
- Domain-Adversarial Neural Networks (**DANN**)
- Adversarial Discriminative Domain Adaptation (**ADDA**)
- Residual Transfer Networks (**RTN**)
- Selective Adversarial Network (**SAN**)
- Importance Weighted Adversarial Network (**IWAN**)
- Partial Adversarial Domain Adaptation (**PADA**)

Table 1. Classification Accuracy (%) for Partial Domain Adaptation on Office-Home Dataset (**ResNet**)

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 [15]	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
DANN [10]	43.76	67.90	77.47	63.73	58.99	67.59	56.84	37.07	76.37	69.15	44.30	77.48	61.72
ADDA [37]	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
RTN [22]	49.31	57.70	**80.07**	63.54	63.47	73.38	65.11	41.73	75.32	63.18	43.57	80.50	63.07
IWAN [43]	53.94	54.45	78.12	61.31	47.95	63.32	54.17	52.02	81.28	**76.46**	56.75	82.90	63.56
SAN [5]	44.42	68.68	74.60	**67.49**	64.99	**77.80**	59.78	44.72	80.07	72.18	50.21	78.66	65.30
PADA [6]	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
ETN	**59.24**	**77.03**	79.54	62.92	**65.73**	75.01	**68.29**	**55.37**	**84.37**	75.72	**57.66**	**84.54**	**70.45**

Table 2. Classification Accuracy (%) for Partial Domain Adaptation on Office-31 and ImageNet-Caltech (**ResNet**)

Method	Office-31							ImageNet-Caltech		Avg
	A→W	D→W	W→D	A→D	D→A	W→A	Avg	I → C	C → I	
ResNet [15]	75.59±1.09	96.27±0.85	98.09±0.74	83.44±1.12	83.92±0.95	84.97±0.86	87.05±0.94	69.69±0.78	71.29±0.74	70.49±0.76
DAN [21]	59.32±0.49	73.90±0.38	90.45±0.36	61.78±0.56	74.95±0.67	67.64±0.29	71.34±0.46	71.30±0.46	60.13±0.50	65.72±0.48
DANN [10]	73.56±0.15	96.27±0.26	98.73±0.20	81.53±0.23	82.78±0.18	86.12±0.15	86.50±0.20	70.80±0.66	67.71±0.76	69.23±0.71
ADDA [37]	75.67± 0.17	95.38±0.23	99.85±0.12	83.41± 0.17	83.62±0.14	84.25±0.13	87.03±0.16	71.82±0.45	69.32±0.41	70.57±0.43
RTN [22]	78.98±0.55	93.22±0.52	85.35±0.47	77.07±0.49	89.25±0.39	89.46±0.37	85.56±0.47	75.50±0.29	66.21±0.31	70.85±0.30
IWAN [43]	89.15±0.37	99.32±0.32	99.36±0.24	90.45±0.36	95.62±0.29	94.26±0.25	94.69±0.31	78.06±0.40	73.33±0.46	75.70±0.43
SAN [5]	93.90±0.45	99.32±0.52	99.36±0.12	94.27±0.28	94.15±0.36	88.73±0.44	94.96±0.36	77.75±0.36	**75.26**±0.42	76.51±0.39
PADA [6]	86.54±0.31	99.32±0.45	**100.00**±.00	82.17±0.37	92.69±0.29	**95.41**±0.33	92.69±0.29	75.03±0.36	70.48±0.44	72.76±0.40
ETN	**94.52**±0.20	**100.00**±.00	**100.00**±.00	**95.03**±0.22	**96.21**±0.27	94.64±0.24	**96.73**±0.16	**83.23**±0.24	74.93±0.28	**79.08**±0.26

Ешперимент

Table 3. Classification Accuracy (%) of ETN and its variants for Partial Domain Adaptation on Office-Home Dataset (**ResNet**)

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
ETN w/o classifier	56.18	71.93	79.32	**65.11**	65.57	73.66	65.47	52.90	82.88	72.93	56.93	82.91	68.93
ETN w/o auxiliary	48.36	50.42	79.13	56.57	45.88	65.49	56.38	49.07	77.53	75.57	**58.81**	78.32	61.79
ETN	**59.24**	**77.03**	**79.54**	62.92	**65.73**	**75.01**	**68.29**	**55.37**	**84.37**	**75.72**	57.66	**84.54**	**70.45**

w/o classifier: without weights on the source classifier

Table 4. Classification Accuracy (%) for Partial Domain Adaptation on Office-31 (**VGG**)

Method	Office-31						
	A→W	D→W	W→D	A→D	D→A	W→A	Avg
VGG [34]	60.34±0.84	97.97±0.63	99.36±0.36	76.43±0.48	72.96±0.56	79.12±0.54	81.03± 0.57
DAN [21]	58.78±0.43	85.86±0.32	92.78±0.28	54.76±0.44	55.42±0.56	67.29±0.20	69.15±0.37
DANN [10]	50.85±0.12	95.23±0.24	94.27±0.16	57.96±0.20	51.77±0.14	62.32±0.12	68.73±0.16
ADDA [37]	53.28±0.15	94.33±0.18	95.36±0.08	58.78±0.12	50.24±0.10	63.34±0.08	69.22±0.12
RTN [22]	69.35±0.42	98.42±0.48	99.59±0.32	75.43±0.38	81.45±0.32	82.98±0.36	84.54±0.38
IWAN [43]	82.90±0.31	79.75±0.26	88.53±0.16	**90.95**±0.33	89.57±0.24	93.36±0.22	87.51±0.25
SAN [5]	83.39±0.36	99.32±0.45	**100.00**±.00	90.70±0.20	87.16±0.23	91.85±0.35	92.07±0.27
PADA [6]	**86.05**±0.36	99.42±0.24	**100.00**±.00	81.73±0.34	93.00±0.24	**95.26**±0.27	92.54±0.24
ETN	85.66±0.16	**100.00**±.00	**100.00**±.00	89.43±0.17	**95.93**±0.23	92.28±0.20	**96.74**±0.13

Ешперимент

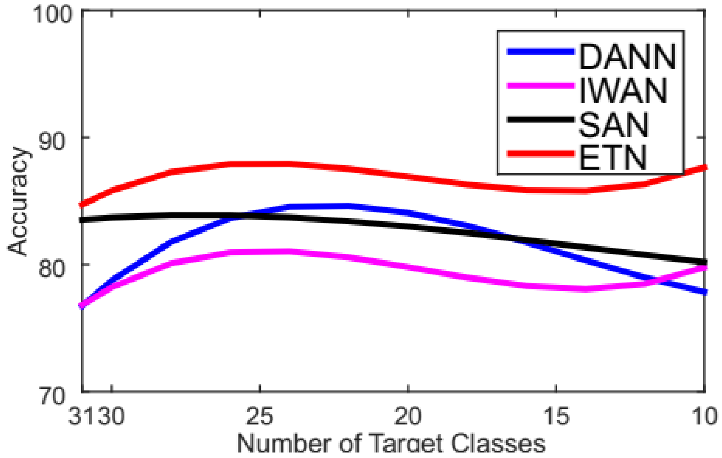


Figure 4. Accuracy by varying #target classes.

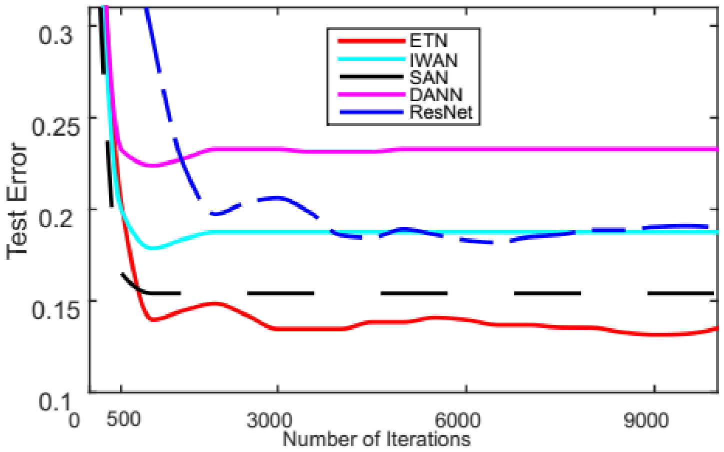
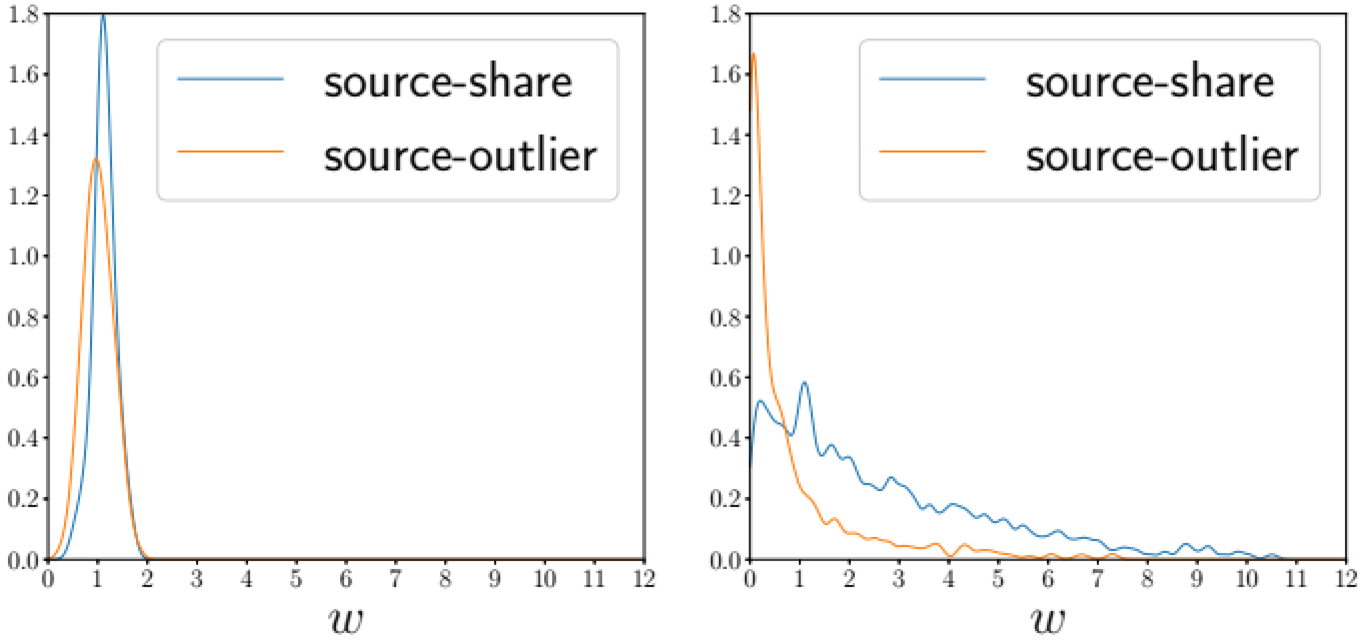


Figure 5. Target test error w.r.t. to #iterations.



(a) IWAN (b) ETN

Figure 6. Density function of the importance weights of source examples in the shared label space $\mathcal{C}_t$ and outlier label space $\mathcal{C}_s \backslash \mathcal{C}_t$ for IWAN and ETN.