Domain adaptation

王曦

Transfer learning

- **利用了已有的知识、模型、结构来帮助我们达成在目标数据上的学习目标**

- **Pre-train and fine-tune**

Domain adaptation

- “A domain is defined by a corpus from a specific source, and may differ from other domains in topic, genre, style, level of formality, etc.” (Koehn and Knowles, 2017).

- Source domain → Target domain

- distribution shift/gap

Settings

- Unsupervised Domain Adaptation
- Semi-supervised Domain Adaptation
- Universal Domain Adaptation (open-set/partial)
- Source-Free Domain Adaptation

Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation

Jian Liang [1] Dapeng Hu [1] Jiashi Feng [1]

ICML 2020

Motivation

- Deep neural networks have achieved remarkable success in a variety of applications across different fields but at the expense of laborious large-scale training data annotation. Transferring a light trained model is much efficient than heavy data transmission.

- Nowadays, data are distributed on different devices and usually contain private information, e.g., those on personal phones or from surveillance cameras. Existing DA methods need to access the source data during learning to adapt, which is not efficient for data transmission and may violate the data privacy policy.

Introduction

- We propose a simple yet generic solution called **Source HypOthesis Transfer (SHOT)**. SHOT assumes that the same deep neural network model consists of a feature encoding module and a classifier module (hypothesis) across domains.
- SHOT freezes the source hypothesis and fine-tunes the source encoding module by maximizing the mutual information between intermediate feature representations and outputs of the classifier, as **information maximization** encourages the network to assign disparate one-hot encodings to the target feature representations.
- We propose a novel self-supervised **pseudo-labeling method** to augment the target representation learning.
- Considering pseudo labels generated by the source classifier may be inaccurate and noisy for target data, we propose to attain **intermediate class-wise prototypes** for the target domain itself and further obtain cleaner pseudo labels via supervision from these prototypes to guide the mapping module learning.

Method

- Source Model Generation

We consider to develop a deep neural network and learn the source model $f_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ by minimizing the following standard cross-entropy loss,

$$\mathcal{L}_{src}(f_s; \mathcal{X}_s, \mathcal{Y}_s) = \\ -\mathbb{E}_{(x_s, y_s) \in \mathcal{X}_s \times \mathcal{Y}_s} \sum_{k=1}^{K} q_k \log \delta_k(f_s(x_s)), \tag{1}$$

where $\delta_k(a) = \frac{\exp(a_k)}{\sum_i \exp(a_i)}$ denotes the k-th element in the soft-max output of a K-dimensional vector a, and q is the one-of-K encoding of y_s where q_k is '1' for the correct class and '0' for the rest. To further increase the discrim-

Method

- Source Model Generation

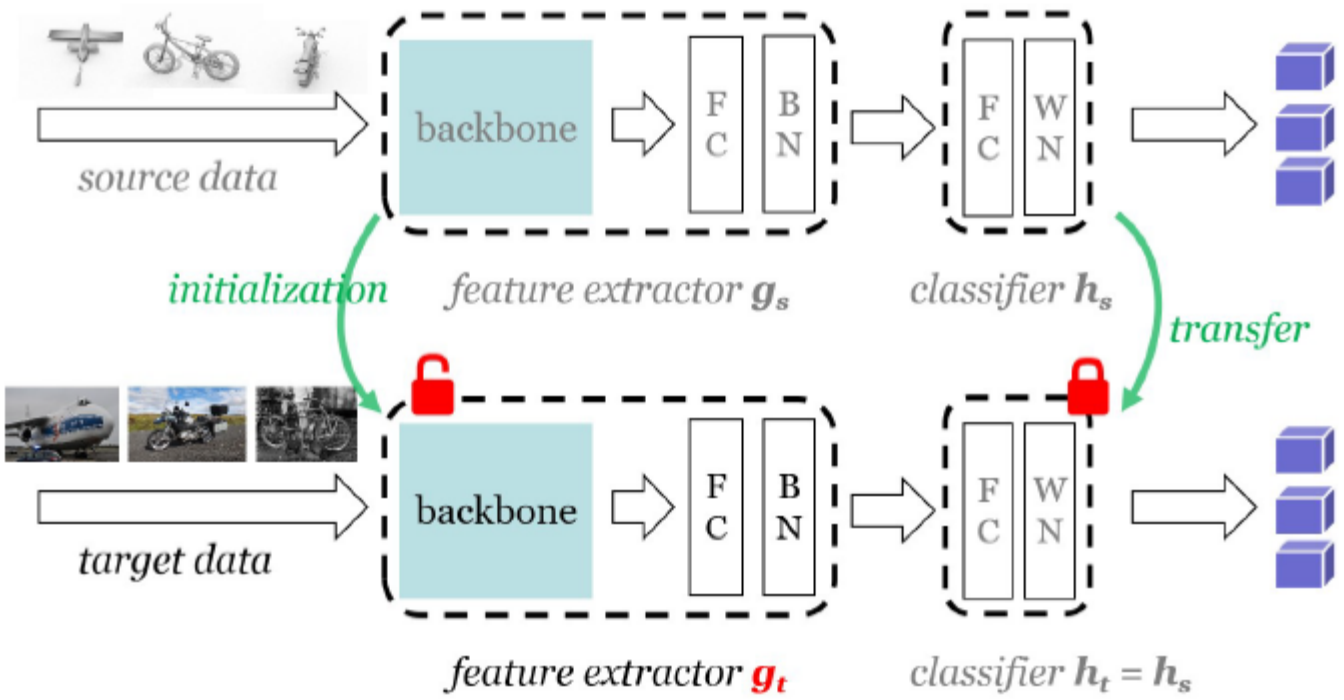
class and '0' for the rest. To further increase the discriminability of the source model and facilitate the following target data alignment, we propose to adopt the label smoothing (LS) technique as it encourages examples to lie in tight evenly separated clusters (Müller et al., 2019). With LS, the objective function is changed to

$$\mathcal{L}_{src}^{ls}(f_s; \mathcal{X}_s, \mathcal{Y}_s) = \\ -\mathbb{E}_{(x_s,y_s)\in\mathcal{X}_s\times\mathcal{Y}_s} \sum_{k=1}^{K} q_k^{ls} \log \delta_k(f_s(x_s)), \tag{2}$$

where $q_k^{ls} = (1-\alpha)q_k + \alpha/K$ is the smoothed label and α is the smoothing parameter which is empirically set to 0.1.

Method

- Source Hypothesis Transfer with Information Maximization (SHOT-IM)



Method

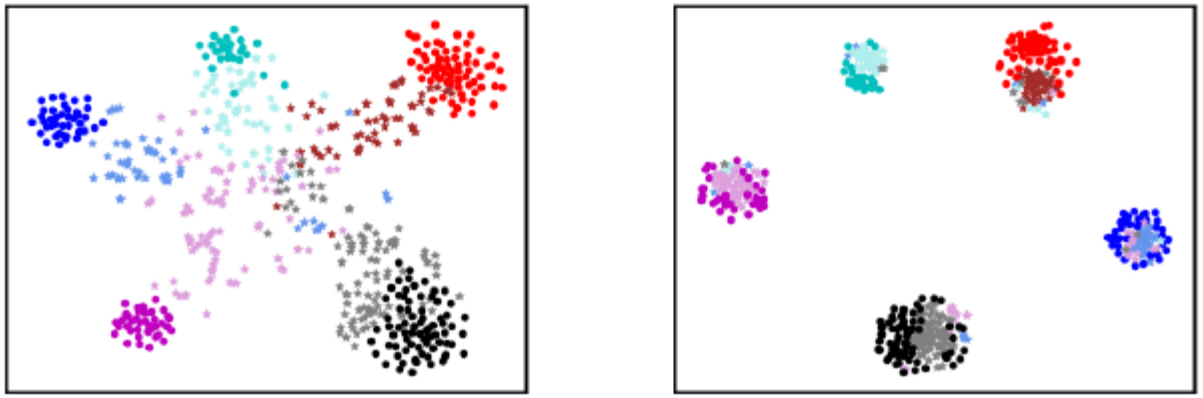
- Source Hypothesis Transfer with Information Maximization (SHOT-IM)

encoding but differ from each other. For this purpose, we adopt the information maximization (IM) loss (Krause et al., 2010; Shi & Sha, 2012; Hu et al., 2017), to make the target outputs individually certain and globally diverse. In practice, we minimize the following $\mathcal{L}_{ent}$ and $\mathcal{L}_{div}$ that together constitute the IM loss:

$$\begin{aligned}
\mathcal{L}_{ent}(f_t; \mathcal{X}_t) &= -\mathbb{E}_{x_t \in \mathcal{X}_t} \sum_{k=1}^{K} \delta_k(f_t(x_t)) \log \delta_k(f_t(x_t)), \\
\mathcal{L}_{div}(f_t; \mathcal{X}_t) &= \sum_{k=1}^{K} \hat{p}_k \log \hat{p}_k \\
&= D_{KL}(\hat{p}, \frac{1}{K}\mathbf{1}_K) - \log K,
\end{aligned} \tag{3}$$

Method

- Source Hypothesis Transfer Augmented with Self-supervised Pseudo-labeling



(a) Source model only (b) SHOT-IM

Figure 1. t-SNE visualizations for a 5-way classification task. Circles in dark colors denote unseen source data and stars in light denote target data. Different colors represent different classes.

Method

- Source Hypothesis Transfer Augmented with Self-supervised Pseudo-labeling

pseudo labels that are conventionally generated by source hypotheses are still noisy due to domain shift. Inspired by DeepCluster (Caron et al., 2018), we propose a novel self-supervised pseudo-labeling strategy. First, we attain the centroid for each class in the target domain, similar to weighted k-means clustering,

$$c_k^{(0)} = \frac{\sum_{x_t \in \mathcal{X}_t} \delta_k(\hat{f}_t(x_t))\, \hat{g}_t(x_t)}{\sum_{x_t \in \mathcal{X}_t} \delta_k(\hat{f}_t(x_t))}, \tag{4}$$

where $\hat{f}_t = \hat{g}_t \circ h_t$ denotes the previously learned target hypothesis. These centroids can robustly and more reliably

Method

- Source Hypothesis Transfer Augmented with Self-supervised Pseudo-labeling

hypothesis. These centroids can robustly and more reliably characterize the distribution of different categories within the target domain. Then, we obtain the pseudo labels via the nearest centroid classifier:

$$\hat{y}_t = \arg\min_k D_f(\hat{g}_t(x_t), c_k^{(0)}), \tag{5}$$

where $D_f(a, b)$ measures the cosine distance between a and b. Finally, we compute the target centroids based on the new pseudo labels:

$$\begin{aligned} c_k^{(1)} &= \frac{\sum_{x_t \in \mathcal{X}_t} \mathbb{1}(\hat{y}_t = k)\, \hat{g}_t(x_t)}{\sum_{x_t \in \mathcal{X}_t} \mathbb{1}(\hat{y}_t = k)}, \\ \hat{y}_t &= \arg\min_k D_f(\hat{g}_t(x_t), c_k^{(1)}). \end{aligned} \tag{6}$$

We term $\hat{y}_t$ as self-supervised pseudo labels since they are generated by the centriods obtained in an unsupervised manner. In practice, we update the centroids and labels in Eq. (6) for multiple rounds. However, experiments verify that updating for once gives sufficiently good pseudo labels.

Method

- To summarize, given the source model fs = gs·hs and pseudo labels generated as above, SHOT freezes the hypothesis from source ht = hs and learns the feature encoder gt with the full objective as:

$$\mathcal{L}(g_t) = \mathcal{L}_{ent}(h_t \circ g_t; \mathcal{X}_t) + \mathcal{L}_{div}(h_t \circ g_t; \mathcal{X}_t) - \beta\, \mathbb{E}_{(x_t, \hat{y}_t) \in \mathcal{X}_t \times \hat{\mathcal{Y}}_t} \sum_{k=1}^{K} \mathbb{1}_{[k=\hat{y}_t]} \log \delta_k(h_t(g_t(x_t))), \quad (7)$$

where $\beta > 0$ is a balancing hyper-parameter.

Experiments

Benchmark

- Office
- Office-Home
- VisDA-C
- Digits

Experiments

Baseline Methods

- **vanilla unsupervised DA in digit recognition**

ADDA, ADR, CDAN, CyCADA, CAT and SWD.

- **object recognition**

DANN, DAN, ADR, CDAN, CAT, SAFN, BSP, TransNorm and SWD

- **specific partial-set and open-set DA tasks**

IWAN, SAN, ETN, SAFN, ATI-λ, OSBP, OpenMax and STA.

- **mutli-source and multi-target DA**

DCTN, MCD, M3SDA-β, FADA, DANN and DADA.

Experiments

Table 2. Classification accuracies (%) on **Digits** dataset for *vanilla closed-set DA*. S: SVHN, M:MNIST, U: USPS.

Method (Source→Target)	S→M	U→M	M→U	Avg.
Source only (Hoffman et al., 2018)	67.1±0.6	69.6±3.8	82.2±0.8	73.0
ADDA (Tzeng et al., 2017)	76.0±1.8	90.1±0.8	89.4±0.2	85.2
ADR (Saito et al., 2018a)	95.0±1.9	93.1±1.3	93.2±2.5	93.8
CDAN+E (Long et al., 2018)	89.2	98.0	95.6	94.3
CyCADA (Hoffman et al., 2018)	90.4±0.4	96.5±0.1	95.6±0.4	94.2
rRevGrad+CAT (Deng et al., 2019)	98.8±0.0	96.0±0.9	94.0±0.7	96.3
SWD (Lee et al., 2019a)	98.9±0.1	97.1±0.1	**98.1**±0.1	98.0
Source model only	70.2±1.2	88.0±2.2	79.7±2.5	79.3
SHOT-IM (ours)	**99.0**±0.1	97.6±0.5	97.7±0.1	98.2
SHOT (full, ours)	98.9±0.1	**98.0**±0.6	97.9±0.2	**98.3**
Target-supervised (Oracle)	99.4±0.1	99.4±0.1	98.0±0.1	98.8

Experiments

Table 3. Classification accuracies (%) on small-sized **Office** dataset for *vanilla closed-set DA* (ResNet-50).

Method (Source→Target)	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
ResNet-50 (He et al., 2016)	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DANN (Ganin & Lempitsky, 2015)	79.7	82.0	68.2	96.9	67.4	99.1	82.2
DAN (Long et al., 2015)	78.6	80.5	63.6	97.1	62.8	99.6	80.4
CDAN+E (Long et al., 2018)	92.9	94.1	71.0	98.6	69.3	**100.**	87.7
rRevGrad+CAT (Deng et al., 2019)	90.8	94.4	72.2	98.0	70.2	**100.**	87.6
SAFN+ENT (Xu et al., 2019)	90.7	90.1	73.0	98.6	70.2	99.8	87.1
CDAN+BSP (Chen et al., 2019)	93.0	93.3	73.6	98.2	72.6	**100.**	88.5
CDAN+TransNorm (Wang et al., 2019)	**94.0**	**95.7**	73.4	**98.7**	74.2	**100.**	**89.3**
Source model only	80.8	76.9	60.3	95.3	63.6	98.7	79.3
SHOT-IM (ours)	90.6	91.2	72.5	98.3	71.4	99.9	87.3
SHOT (full, ours)	**94.0**	90.1	**74.7**	98.4	**74.3**	99.9	88.6

Office (Saenko et al., 2010) is a standard DA benchmark which contains three domains (Amazon (**A**), DSLR (**D**), and Webcam (**W**)) and each domain consists of 31 object classes under the office environment. (Gong et al., 2012) further extracts 10 shared categories between **Office** and Caltech-256 (**C**), and forms a new benchmark named **Office-Caltech**. Both benchmarks are small-sized.

Experiments

Table 4. Classification accuracies (%) on medium-sized **Office-Home** dataset for *vanilla closed-set DA* (ResNet-50).

Method (Source→Target)	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg.
ResNet-50 (He et al., 2016)	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 (Ganin & Lempitsky, 2015)	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
DAN (Long et al., 2015)	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
CDAN+E (Long et al., 2018)	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
CDAN+BSP (Chen et al., 2019)	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	**59.3**	81.9	66.3
SAFN (Xu et al., 2019)	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
CDAN+TransNorm (Wang et al., 2019)	50.2	71.4	77.4	59.3	72.7	73.1	61.0	53.1	79.5	71.9	59.0	82.9	67.6
Source model only	44.6	67.3	74.8	52.7	62.7	64.8	53.0	40.6	73.2	65.3	45.4	78.0	60.2
SHOT-IM (ours)	55.4	76.6	80.4	66.9	74.3	75.4	65.6	54.8	80.7	**73.7**	58.4	83.4	70.5
SHOT (full, ours)	**57.1**	**78.1**	**81.5**	**68.0**	**78.2**	**78.1**	**67.4**	**54.9**	**82.2**	73.3	58.8	**84.3**	**71.8**

Office-Home (Venkateswara et al., 2017) is a challenging medium-sized benchmark, which consists of four distinct domains (Artistic images (**Ar**), Clip Art (**Cl**), Product images (**Pr**), and Real-World images (Rw)). There are totally 65 everyday objects categories in each domain.

Experiments

Table 5. Classification accuracies (%) on large-scale **VisDA-C** dataset for *vanilla closed-set DA* (ResNet-101).

Method (Synthesis → Real)	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
ResNet-101 (He et al., 2016)	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN (Ganin & Lempitsky, 2015)	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN (Long et al., 2015)	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
ADR (Saito et al., 2018a)	94.2	48.5	84.0	**72.9**	90.1	74.2	**92.6**	72.5	80.8	61.8	82.2	28.8	73.5
CDAN (Long et al., 2018)	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP (Chen et al., 2019)	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN (Xu et al., 2019)	93.6	61.3	**84.1**	70.6	**94.1**	79.0	91.8	79.6	89.9	55.6	**89.0**	24.4	76.1
SWD (Lee et al., 2019a)	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
Source model only	60.9	21.6	50.9	67.6	65.8	6.3	82.2	23.2	57.3	30.6	84.6	8.0	46.6
SHOT-IM (ours)	93.7	86.4	78.7	50.7	91.0	93.5	79.0	78.3	89.2	85.4	87.9	51.1	80.4
SHOT (full, ours)	**94.3**	**88.5**	80.1	57.3	93.1	**94.9**	80.7	**80.3**	**91.5**	**89.1**	86.3	**58.2**	**82.9**

VisDA-C (Peng et al., 2017) is a challenging large-scale benchmark that mainly focuses on the 12-class synthesis-to-real object recognition task. The source domain contains 152 thousand synthetic images generated by rendering 3D models while the target domain has 55 thousand real object images sampled from Microsoft COCO.

Experiments

Ablation study

Table 6. Average accuracies on three closed-set UDA datasets.

Methods / Datasets	Office	Office-Home	VisDA-C
Source model only	79.3	60.2	46.6
naive pseudo-labeling (PL) (Lee, 2013)	83.0	64.1	76.6
Self-supervised PL (ours)	87.6	68.9	80.7
$\mathcal{L}_{ent}$	83.5	55.5	63.3
$\mathcal{L}_{ent} + \mathcal{L}_{div}$	87.3	70.5	80.4
$\mathcal{L}_{ent} + \mathcal{L}_{div}$ + naive PL (Lee, 2013)	87.5	70.3	82.9
$\mathcal{L}_{ent} + \mathcal{L}_{div}$ + Self-supervised PL	88.6	71.8	82.9

Experiments

Ablation study

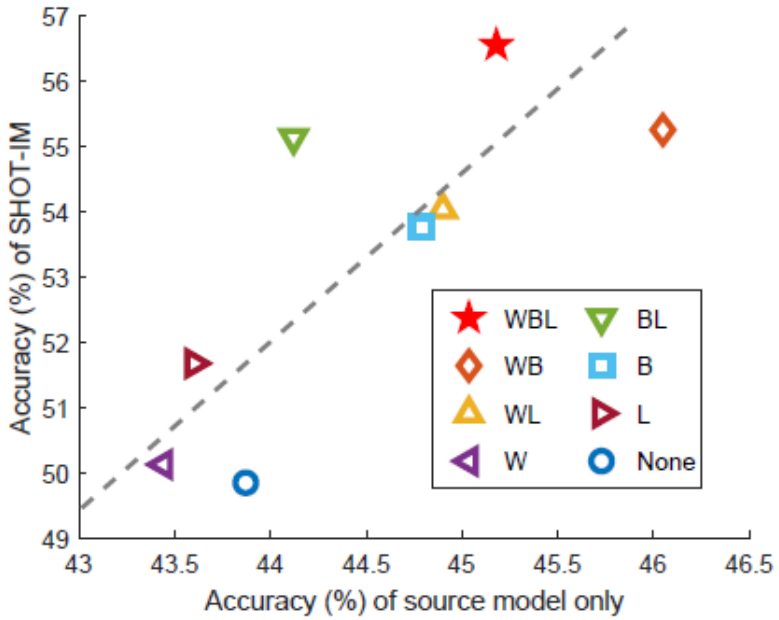


Figure 3. Accuracies (%) on the Ar→Cl task for *closed-set UDA*. [**W**eight normalization/ **B**atch normalization/ **L**abel smoothing]

Experiments

Table 7. (OS) Classification accuracies (%) on **Office-Home** dataset for *partial-set and open-set DA* (ResNet-50).

Partial-set DA (Source→Target)	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg.
ResNet-50 (He et al., 2016)	46.3	67.5	75.9	59.1	59.9	62.7	58.2	41.8	74.9	67.4	48.2	74.2	61.3
IWAN (Zhang et al., 2018a)	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6
SAN (Cao et al., 2018)	44.4	68.7	74.6	67.5	65.0	77.8	59.8	44.7	80.1	72.2	50.2	78.7	65.3
ETN (Cao et al., 2019)	59.2	77.0	79.5	62.9	65.7	75.0	68.3	55.4	84.4	75.7	57.7	84.5	70.5
SAFN (Xu et al., 2019)	58.9	76.3	81.4	70.4	73.0	77.8	72.4	55.3	80.4	75.8	60.4	79.9	71.8
Source model only	45.2	70.4	81.0	56.2	60.8	66.2	60.9	40.1	76.2	70.8	48.5	77.3	62.8
SHOT-IM (ours)	57.9	83.6	88.8	72.4	74.0	79.0	76.1	60.6	**90.1**	**81.9**	**68.3**	**88.5**	76.8
SHOT (full, ours)	**64.8**	**85.2**	**92.7**	**76.3**	**77.6**	**88.8**	**79.7**	**64.3**	89.5	80.6	66.4	85.8	**79.3**

Open-set DA (Source→Target)	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg.
ResNet (He et al., 2016)	53.4	52.7	51.9	69.3	61.8	74.1	61.4	64.0	70.0	78.7	71.0	74.9	65.3
ATI-λ (Panareda Busto & Gall, 2017)	55.2	52.6	53.5	69.1	63.5	74.1	61.7	64.5	70.7	79.2	72.9	75.8	66.1
OSBP (Saito et al., 2018c)	56.7	51.5	49.2	67.5	65.5	74.0	62.5	64.8	69.3	80.6	74.7	71.5	65.7
OpenMax (Bendale & Boult, 2016)	56.5	52.9	53.7	69.1	64.8	74.5	64.1	64.0	71.2	80.3	73.0	76.9	66.7
STA (Liu et al., 2019)	58.1	53.1	54.4	**71.6**	69.3	**81.9**	63.4	**65.2**	74.9	**85.0**	**75.8**	80.8	69.5
Source model only	36.3	54.8	69.1	33.8	44.4	49.2	36.8	29.2	56.8	51.4	35.1	62.3	46.6
SHOT-IM (ours)	62.5	77.8	83.9	60.9	73.4	79.4	64.7	58.7	83.1	69.1	62.0	82.1	71.5
SHOT (full, ours)	**64.5**	**80.4**	**84.7**	63.1	**75.4**	81.2	**65.3**	59.3	**83.3**	69.6	64.6	**82.3**	**72.8**

Experiments

Table 8. Classification accuracies (%) on **Office-Caltech** dataset for *multi-source and multi-target DA* (ResNet-101). [*$\mathfrak{R}$ denotes the **rest** three domains except the single source / target.]

Multi-source ($\mathfrak{R} \rightarrow$)	$\mathfrak{R} \rightarrow$A	$\mathfrak{R} \rightarrow$C	$\mathfrak{R} \rightarrow$D	$\mathfrak{R} \rightarrow$W	Avg.
ResNet-101 (He et al., 2016)	88.7	85.4	98.2	99.1	92.9
DAN (Long et al., 2015)	91.6	89.2	99.1	99.5	94.8
DCTN (Xu et al., 2018)	92.7	90.2	99.0	99.4	95.3
MCD (Saito et al., 2018b)	92.1	91.5	99.1	99.5	95.6
M^{3}SDA-β (Peng et al., 2019a)	94.5	92.2	**99.2**	99.5	96.4
FADA (Peng et al., 2020)	84.2	88.7	87.1	88.1	87.1
Source model only	95.4	93.7	98.9	98.3	96.6
SHOT-IM (ours)	96.2	96.1	98.5	99.7	97.6
SHOT (full, ours)	**96.4**	**96.2**	98.5	**99.7**	**97.7**

Multi-target ($\rightarrow \mathfrak{R}$)	A$\rightarrow \mathfrak{R}$	C$\rightarrow \mathfrak{R}$	D$\rightarrow \mathfrak{R}$	W$\rightarrow \mathfrak{R}$	Avg.
ResNet-101 (He et al., 2016)	90.5	94.3	88.7	82.5	89.0
SE (French et al., 2018)	90.3	94.7	88.5	85.3	89.7
MCD (Saito et al., 2018b)	91.7	95.3	89.5	84.3	90.2
DANN (Ganin & Lempitsky, 2015)	91.5	94.3	90.5	86.3	90.7
DADA (Peng et al., 2019b)	92.0	95.1	91.3	93.1	92.9
Source model only	90.7	96.1	90.2	90.9	92.0
SHOT-IM (ours)	95.7	97.2	**96.3**	96.1	96.3
SHOT (full, ours)	**96.2**	**97.3**	**96.3**	**96.2**	**96.5**

Experiments

Special Case. One may wonder whether SHOT works if we cannot train the source model by ourselves. To find the answer, we utilize the most popular off-the-shelf pre-trained ImageNet models ResNet-50 (He et al., 2016) and consider a PDA task (**ImageNet** → **Caltech**) to evaluate the effectiveness of SHOT with the same basic setting as (Cao et al., 2019). Obviously, in Table 9, SHOT still achieves slightly higher accuracy than the state-of-the-art ETN (Cao et al., 2019) even without accessing the source data.

Table 9. Results of a PDA task (**ImageNet** → **Caltech**). †utilizes the training set of ImageNet besides pre-trained ResNet-50 model.

Methods	ResNet-50	ETN†	SHOT-IM (ours)	SHOT (full, ours)
Accuracy	69.7±0.0	83.2±0.2	81.7±0.5	**83.3**±0.1

Thank you for listening!