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

Jian Liang Dapeng Hu Jiashi Feng

Learning and Vision Lab, National University of Singapore (NUS)

International Conference on Machine Learning, 2020

Outline

Situation 1

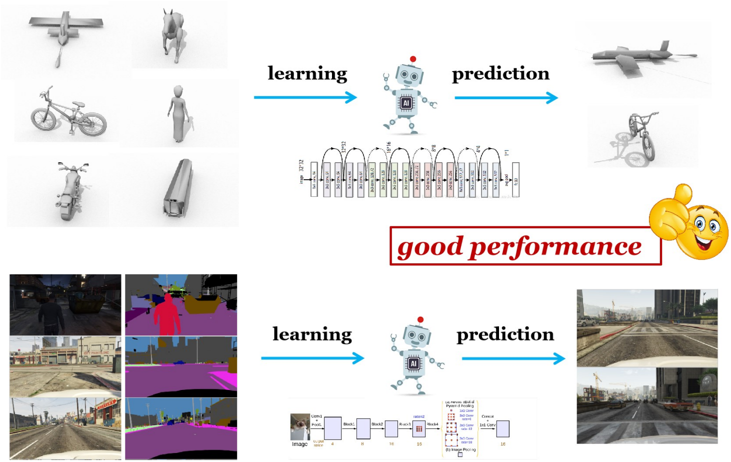


Figure 1: Test data comes from the same distribution as training data!

Situation 2

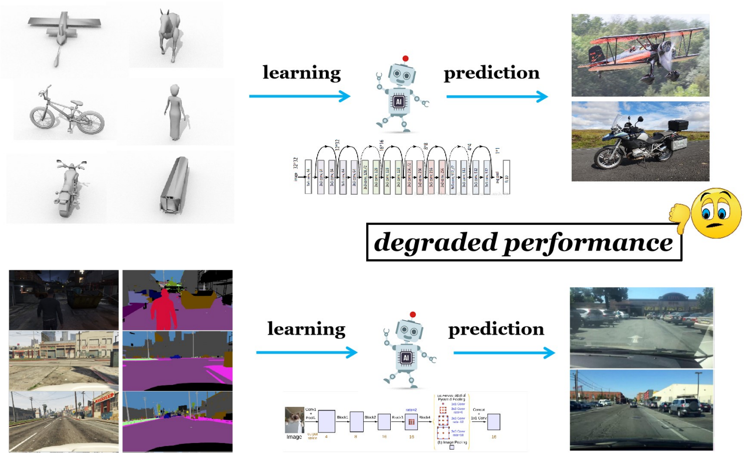


Figure 2: Test data and training data comes from different distributions!

Unsupervised Domain Adaptation (DA)

Vanilla setting

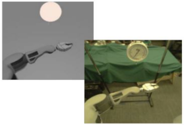
- **Source Domain** $\mathcal{D}_s$: n_s labeled samples $\{x_s^i, y_s^i\}_{i=1}^{n_s}$ from $P_S(X, Y)$;
- **Target Domain** $\mathcal{D}_t$: n_t unlabeled samples $\{x_t^i, ?\}_{i=1}^{n_t}$ from $P_T(X, Y)$;
- **Goal**: Use $\{x_t^i\}_{i=1}^{n_t}$ during training (transductive) and learn a good classifier to get the values of ? under **domain shift** (i.e., $P_S \neq P_T$).

Classification

Detection

Segmentation

Re-identification



Control

Visual Localization

Credit to Gabriela Csurka, TaskCV-2019 talk.

Previous DA Methods

(I) Input-level Pixel Transfer

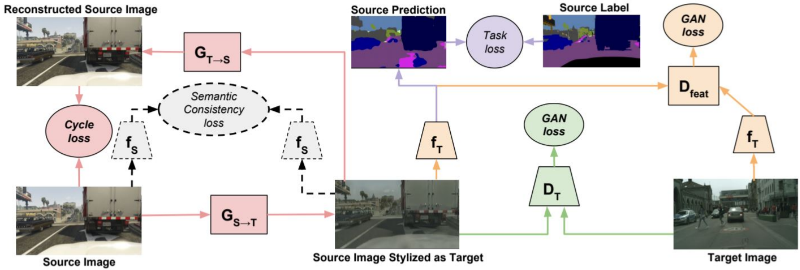
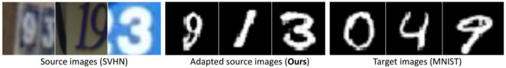


Figure 3: Cycle-consistent adversarial adaptation (CyCADA) [1] overview.



Accuracy on target	
Source-only:	67.1%
Adapted (**ours**):	**90.4%**

[1] Hoffman, Judy, et al. "CyCADA: Cycle-Consistent Adversarial Domain Adaptation." In ICML 2018.

Previous DA Methods

(II) Feature-level Alignment

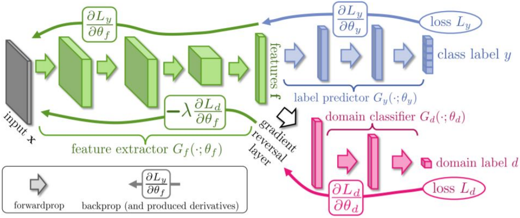


Figure 4: Unsupervised Domain Adaptation by Backpropagation (DANN) [2] overview.

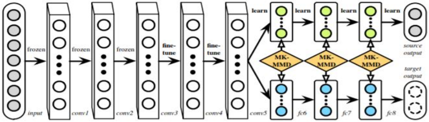


Figure 5: Deep Adaptation Networks (DAN) [3] overview.

[2] Ganin, Yaroslav, and Victor Lempitsky. "Unsupervised Domain Adaptation by Backpropagation." In ICML 2015.

[3] Long, Mingsheng, et al. "Learning Transferable Features with Deep Adaptation Networks." In ICML 2015.

Previous DA Methods

(III) Output-level Regularization

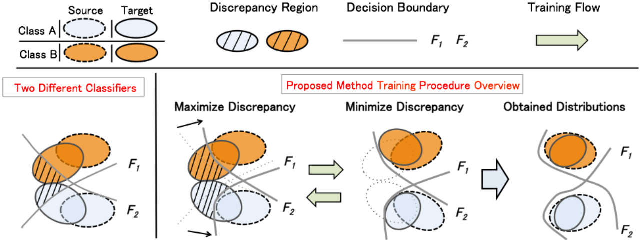


Figure 6: Maximum classifier discrepancy (MCD) [4] overview.

Or exploit the low-density separation principle:

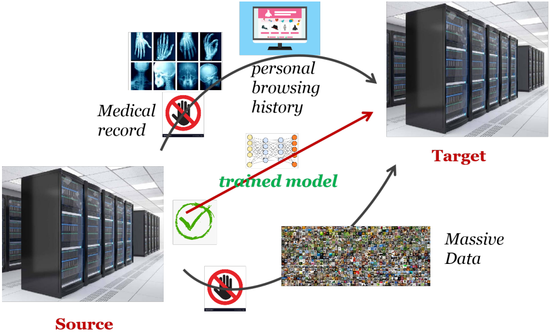
- entropy minimization
- pseudo-labeling / self-training
- virtual adversarial training
- consistency regularization

[4] Saito, Kuniaki, et al. "Maximum classifier discrepancy for unsupervised domain adaptation." In CVPR 2018.

So is DA solved?

Limitation of existing DA methods

- **Not Secure**: the full access to source data is required.
- **Concentrated**: processing different domains in the same machine.



Outline

- **Source Model** $f_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ trained on $\mathcal{D}_s$;
- **Target Domain** $\mathcal{D}_t$: n_t unlabeled samples $\{x_t^i, ?\}_{i=1}^{n_t}$;
- **Goal**: learn a good classifier $f_t : \mathcal{X}_t \rightarrow \mathcal{Y}_t$ to get the values of $?$.

How to generate a good source model f_s?

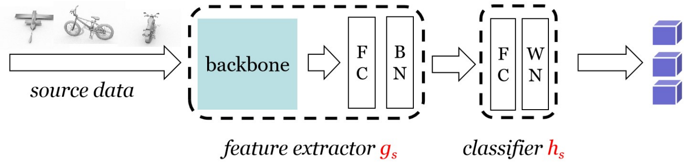


Figure 7: The network of source model for object recognition. $f_s(x) = h_s(g_s(x))$, where $g_s : \mathcal{X}_s \rightarrow \mathbb{R}^d$ and $h_s : \mathbb{R}^d \rightarrow \mathbb{R}^K$.

Classification loss:

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

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.

What we can learn from the model f_s?

$\forall y_s = k$, maximizing $f_s^{(k)}(x_s) = \frac{\exp(w_k^\top g_s(x_s))}{\sum_i \exp(w_i^\top g_s(x_s))}$ means minimizing the distance between $g_s(x_s)$ and w_k, where w_k is the k-th weight vector in h_s.

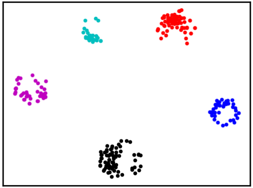


Figure 8: t-SNE visualizations of source features $g_s(x), x \in \mathcal{X}_s$. Each color denotes one class.

Fortunately, even we have no access to the source data or features directly, we may still estimate the distribution of source features via h_s.

Source Hypothesis Transfer

Framework

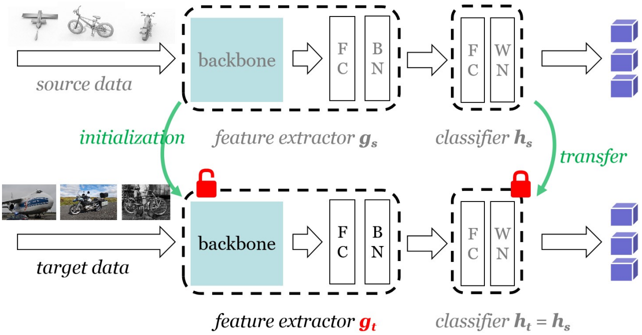


Figure 9: The proposed Source Hypothesis Transfer (SHOT) framework.

Ideally, we expect the feature extractor g_t can produce source-like features for target data, that is to say, the corresponding outputs of h_s are also close to one-hot encoding like those of source features.

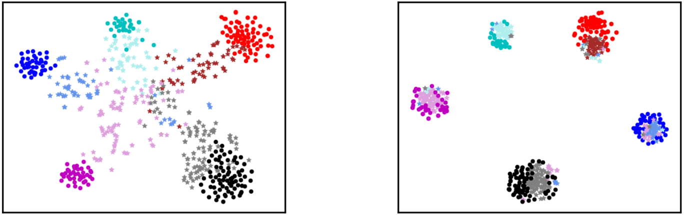
In practice, we minimize the following $\mathcal{L}_{ent}$ and $\mathcal{L}_{div}$ that together constitute the IM loss: $[f_t(x) = h_s(g_t(x))]$

$$\begin{aligned}\mathcal{L}_{ent}(f_t; \mathcal{X}_t) &= -\mathbb{E}_{x_t \in \mathcal{X}_t} \sum_{k=1}^{K} \delta_k(f_t(x_t)) \log \left(\delta_k(f_t(x_t))\right), \\ \mathcal{L}_{div}(f_t; \mathcal{X}_t) &= \sum_{k=1}^{K} \hat{p}_k \log \hat{p}_k = D_{KL}(\hat{p}_k, \frac{1}{K}\mathbf{1}_K) - \log K,\end{aligned} \tag{2}$$

where $f_t(x) = h_t(g_t(x))$ is the K-dimensional output of each target sample, $\hat{p} = \mathbb{E}_{x_t \in \mathcal{X}_t}[\delta(f_t^{(k)}(x_t))]$ is the mean output embedding of the whole target domain, and $\mathbf{1}_K$ is a K-dimensional vector with all ones.

Source Hypothesis Transfer

Information Maximization (IM) - (Cont'd)



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

Figure 10: t-SNE visualizations. Circles in dark colors denote the **unseen** source data and stars in light denote the target data. Different colors represent different classes.

IM loss relies heavily on the initialization and does not fully consider **the structure of target data**. Even features from different domains are well aligned, there still exists **cross-label matching**.

We exploit **target-specific centroids** to obtain accurate pseudo labels.

1. Centroid Initialization & Cluster Assignment.

$$c_k^{(0)} = \frac{\sum_{x_t \in \mathcal{X}_t} \delta(\hat{f}_t^{(k)}(x)) \ \hat{g}_t(x)}{\sum_{x_t \in \mathcal{X}_t} \delta(\hat{f}_t^{(k)}(x))}, \qquad \hat{y}_t = \arg\min_k D_f(\hat{g}_t(x_t), c_k^{(0)}), \tag{3}$$

2. Centroid Update & Cluster Assignment.

$$c_k^{(1)} = \frac{\sum_{x_t \in \mathcal{X}_t} \mathbb{1}(\hat{y}_t = k) \ \hat{g}_t(x)}{\sum_{x_t \in \mathcal{X}_t} \mathbb{1}(\hat{y}_t = k)}, \qquad \hat{y}_t = \arg\min_k D_f(\hat{g}_t(x_t), c_k^{(1)}). \tag{4}$$

* $D_f(a, b)$ measures the cosine distance between a and b.

Source Hypothesis Transfer

Complete objective

$$\mathcal{L}(g_t) = \mathcal{L}_{ent}(h_s \circ g_t; \mathcal{X}_t) + \mathcal{L}_{div}(h_s \circ g_t; \mathcal{X}_t) - \beta \, \mathbb{E}_{(x_t, \hat{y}_t) \in \mathcal{X}_t \times \hat{\mathcal{Y}}_t} \sum_{k=1}^{K} \mathbb{1}_{[k=\hat{y}_t]} \log \left(\delta_k (h_s(g_t(x_t))) \right). \tag{5}$$

Difference with prior work.

Both TDA[a] and MCS[b] are shallow methods that ignore feature representation learning, deteriorating the performance.
FADA[c] is elegantly designed for multi-source domain adaptation.

[a] Chidlovskii, Boris, Stephane Clinchant, and Gabriela Csurka. "Domain adaptation in the absence of source domain data." In KDD 2016.

[b] Liang, Jian, et al. "Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation." In CVPR 2019.

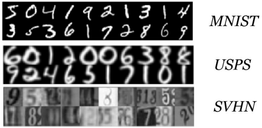
[c] Peng, Xingchao, et al. "Federated Adversarial Domain Adaptation." In ICLR 2020.

Outline

Setup

Data Sets and Various Scenarios

1. Digit recognition (**MNIST**, **USPS**, **SVHN**)
2. Cross-domain object recognition (**Office**, **Office-Home**, **Office-Caltech**)
3. Synthetic-to-real object recognition (**VisDA-C**)



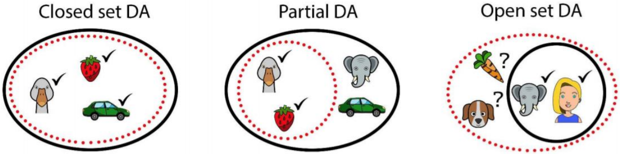


Figure 11: Typical UDA scenarios.

Credit to Marco Toldo et al.

Results

Vanilla Closed-set Domain Adaptation

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.
DANN (ICML 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 (ICML 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 (NeurIPS 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 (ICML 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 (ICCV 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 (NeurIPS 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 (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**
SRDC [5] (CVPR 2020)	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	**85.0**	71.3

Table 1: Accuracies (%) on **Office-Home** dataset (ResNet-50).

Method (Source→Target)	S→M	U→M	M→U	Avg.
ADDA (CVPR 2017)	76.0±1.8	90.1±0.8	89.4±0.2	85.2
ADR (ICLR 2018)	95.0±1.9	93.1±1.3	93.2±2.5	93.8
CDAN+E (NeurIPS 2018)	89.2	98.0	95.6	94.3
CyCADA (ICML 2018)	90.4±0.4	96.5±0.1	95.6±0.4	94.2
rRevGrad+CAT (ICCV 2019)	98.8±0.0	96.0±0.9	94.0±0.7	96.3
SWD (CVPR 2019)	**98.9**±0.1	97.1±0.1	**98.1**±0.1	98.0
Source model only	67.1±0.9	87.8±2.3	89.6±0.4	81.5
SHOT-IM (ours)	89.6±5.0	96.8±0.4	91.9±0.4	92.8
SHOT (ours)	**98.9**±0.0	**98.4**±0.6	98.0±0.2	**98.4**
STAR [6] (CVPR 2020)	98.8±0.1	97.7±0.1	97.8±0.1	98.1
Target-supervised (Oracle)	99.4±0.0	99.4±0.0	98.0±0.1	98.9

Table 2: Accuracies (%) on **Digits** dataset. S: SVHN, M:MNIST, U: USPS.

[5] Tang, Hui, et al. "Unsupervised Domain Adaptation via Structurally Regularized Deep Clustering." In CVPR 2020.

[6] Lu, Zhihe, et al. "Stochastic Classifiers for Unsupervised Domain Adaptation." In CVPR 2020.

Results

Multi-source and Multi-target Domain Adaptation

Multi-source ($\mathbf{R} \rightarrow$)	$\mathbf{R}$→A	$\mathbf{R}$→C	$\mathbf{R}$→D	$\mathbf{R}$→W	Avg.
FADA (ICLR 2020)	84.2	88.7	87.1	88.1	87.1
DAN (ICML 2015)	91.6	89.2	99.1	99.5	94.8
DCTN (CVPR 2018)	92.7	90.2	99.0	99.4	95.3
MCD (CVPR 2018)	92.1	91.5	99.1	99.5	95.6
M^{3}SDA-β (ICCV 2019)	94.5	92.2	**99.2**	99.5	96.4
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 (ours)	**96.4**	**96.2**	98.5	**99.7**	**97.7**
Multi-target ($\rightarrow \mathbf{R}$)	A→$\mathbf{R}$	C→$\mathbf{R}$	D→$\mathbf{R}$	W→$\mathbf{R}$	Avg.
SE (ICLR 2018)	90.3	94.7	88.5	85.3	89.7
MCD (CVPR 2018)	91.7	95.3	89.5	84.3	90.2
DANN (ICML 2015)	91.5	94.3	90.5	86.3	90.7
DADA (ICML 2019)	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 (ours)	**96.2**	**97.3**	**96.3**	**96.2**	**96.5**

Table 3: Accuracies (%) on **Office-Caltech** dataset (ResNet-101). [$^*\mathbf{R}$ denotes the **rest** three domains except the single source / target.]

Results

Partial-set and Open-set Domain Adaptation

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.
IWAN (CVPR 2018)	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 (ECCV 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 (CVPR 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 (ICCV 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	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)	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**
RTNet$_{adv}$ [7] (CVPR 2020)	63.2	80.1	80.7	66.7	69.3	77.2	71.6	53.9	84.6	77.4	57.9	85.5	72.3

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.
ATI-λ (ICCV 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 (ECCV 2018)	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 (CVPR 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 (CVPR 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**
Inheritune [8] (CVPR 2020)	60.1	70.9	83.2	64.0	70.0	75.7	**66.1**	54.2	81.3	74.9	56.2	78.6	69.6

Table 4: Accuracies (%) on **Office-Home** dataset (ResNet-50).

[7] Chen, Zhihong, et al. "Selective transfer with reinforced transfer network for partial domain adaptation." In CVPR 2020.

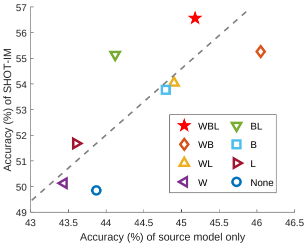
[8] Kundu, Jogendra Nath, et al. "Towards Inheritable Models for Open-Set Domain Adaptation." In CVPR 2020.

Analysis

Ablation study

Methods / Datasets	**Office**	**Office-Home**	**VisDA-C**
Source model only	79.3	60.2	46.6
naive pseudo-labeling (PL)[9]	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	87.5	70.3	82.9
$\mathcal{L}_{ent} + \mathcal{L}_{div}$ + Self-supervised PL	88.6	71.8	82.9

Table 5: Average accuracies on three closed-set UDA datasets.



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

[9] Lee, Dong-Hyun. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." In ICML Workshop 2013.

If we cannot train the source model by ourselves?

An interesting example

To find the answer, we utilize the most popular off-the-shelf pre-trained ImageNet model ResNet-50 and consider a PDA task (**ImageNet** $\rightarrow$ **Caltech**) to evaluate the effectiveness of SHOT below.

Methods	ResNet-50	ETN†[10]	SHOT-IM	SHOT
Accuracy	69.7 $\pm$ 0.0	83.2 $\pm$ 0.2	81.7$\pm$ 0.5	**83.3** $\pm$ 0.1

Table 6: Results of a PDA task (**ImageNet** $\rightarrow$ **Caltech**). †utilizes the training set of ImageNet besides the off-the-shelf pre-trained ResNet-50 model.

[10] Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." In CVPR 2019.

1. With only the source model provided, **our approach** achieves competitive and even state-of-the-art performance.
2. Feature alignment can be achieved **implicitly with output-level regularization** like entropy minimization and information maximization.
3. To combat domain shift, **self-supervision** from the target domain itself is quite critical.

Additional Discussions with Concurrent Works

1. Main Idea of MoA [11]
   - generate pseudo source samples
   - pseudo labeled source data & unlabeled target data (semi-supervised learning)
2. Main Idea of USFDA [12]
   - simulate labeled negative samples
   - entropy minimization with fixed decision boundary
3. Main Idea of SFDA [13]
   - extra target-specific classifier (prototype based) in addition to source-oriented classifier

Difference: We need no additional components like data generator or classifier within the training algorithm.

---

[11] Li, Rui, et al. "Model Adaptation: Unsupervised Domain Adaptation without Source Data." In CVPR 2020.

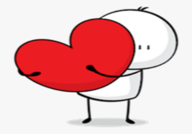
[12] Kundu, Jogendra Nath, Naveen Venkat, and R. Venkatesh Babu. "Universal Source-Free Domain Adaptation." In CVPR 2020.

[13] Kim, Youngeun, et al. "Domain Adaptation without Source Data." Submitted to NeurIPS 2020.

Thank you!

1. Code is available at https://github.com/tim-learn/SHOT/.
2. If you require any further information, feel free to contact me.

Email: liangjian92@gmail.com



(a) Love

(b) Peace

(c) Health