Source Data-absent Unsupervised Domain Adaptation through Hypothesis Transfer and Labeling Transfer — Supplementary Material

Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, *Senior Member, IEEE,* Jiashi Feng, *Member, IEEE*

A. Upper bound of SHOT (w/ source data)

To turn the proposed source data-absent method into a source data-dependent method, we keep the first stage (source model generation) of SHOT unchanged, then directly incorporate the source classification loss in Eq. (2) into the overall loss in Eq. (10). Note that, in the target adaptation stage, the source hypothesis (classifier layer) is no longer fixed, instead, data from both domains are used to learn the domain-shared classifier at the same time.

In this manner, we exploit the source data when training SHOT-IM, SHOT, SHOT-IM++, and SHOT++, respectively, and show the final results for closed-set UDA on Office-Home in Table I. It is easy to discover that with the involvement of source data, all the methods obtain boosted performance in terms of the average accuracy. Besides, the improvement (about 1.1) over SHOT-IM and SHOT-IM++ is larger than that (about 0.5) over SHOT and SHOT++, which indicates that using the proposed self-supervised techniques like pseudo-labeling in the target domain itself plays a similar role as source data.

TABLE I
Classification accuracies (%) on **Office-Home** for *vanilla closed-set UDA*.

Methods	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.
SHOT-IM	55.9	76.8	80.6	66.7	73.7	75.4	65.4	54.9	80.9	73.2	58.5	83.5	70.5
w/ source data	57.5	77.5	81.2	67.4	75.8	77.0	67.0	55.9	82.1	73.5	58.9	84.1	71.5
	+1.6	+0.7	+0.6	+0.7	+2.1	+1.6	+1.6	+1.0	+1.2	+0.3	+0.4	+0.6	+1.0
SHOT-IM++	56.9	77.7	81.5	67.6	74.9	76.9	66.1	55.9	81.7	73.8	59.3	84.4	71.4
w/ source data	58.6	78.7	82.1	68.6	77.4	78.7	67.9	57.0	83.2	74.0	59.8	84.9	72.6
	+1.7	+1.0	+0.6	+1.0	+2.5	+1.8	+1.8	+1.1	+1.5	+0.2	+0.5	+0.5	+1.2
SHOT	57.7	79.1	81.5	67.6	77.9	77.8	68.1	55.8	82.0	72.8	59.7	84.4	72.0
w/ source data	57.7	79.2	82.2	68.0	79.4	78.3	67.8	56.7	82.6	73.2	60.1	84.9	72.5
	–	+0.1	+0.7	+0.4	+1.5	+0.5	-0.3	+0.9	+0.6	+0.4	+0.4	+0.5	+0.5
SHOT++	57.9	79.7	82.5	68.5	79.6	79.3	68.5	57.0	83.0	73.7	60.7	84.9	73.0
w/ source data	58.6	80.0	83.2	69.2	80.7	79.9	68.4	57.5	83.6	73.7	61.2	85.4	73.5
	+0.7	+0.3	+0.7	+0.7	+1.1	+0.6	-0.1	+0.5	+0.6	–	+0.5	+0.5	+0.5

B. Sensitivity of the IM loss

To investigate the sensitivity of the diversity term $\mathcal{L}_{div}$ within the IM loss in Eq.(3), we conduct an experimental study and show the results in Fig. 1 and Table II. As shown in Fig. 1, we rank 12 classes based on the number of training samples in the target domain (i.e. validation set of the VISDA-C dataset), denoted as 's0'. To vary the degree of class imbalance in the training target domain, we first drop half of the samples in the largest 6 classes, denoted as 's1'. Then we drop half of the samples in the smallest 6 classes, denoted as 's2'. Besides, we make the number of training samples of each class as the smallest class 'knife', forming 's3'. Generally, 's2' has the largest class imbalance, and 's3' has the smallest class imbalance, and the class imbalance degree of 's1' is smaller than that of 's0'. For all four settings, we use the whole real-world dataset (i.e., the original validation set of VISDA-C) as the target test set.

To verify the effectiveness and sensitivity of the diversity term within the IM loss, we conduct comparison between SHOT-IM ($\beta = 0$) and SHOT-IM on four aforementioned settings and show the results in Table II. It is easy to find that SHOT-IM outperforms SHOT-IM ($\beta = 0$) in terms of the per-class accuracy for all four settings. Besides, with the increasing degree of class imbalance (s3→s1→s0→s2), the per-class accuracies of SHOT-IM and SHOT-IM ($\beta = 0$) decrease, and the accuracy of

J. Liang is with National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, 100190. E-mail: liangjian92@gmail.com.
Manuscript received October 22, 2020; revised May 28, 2021.

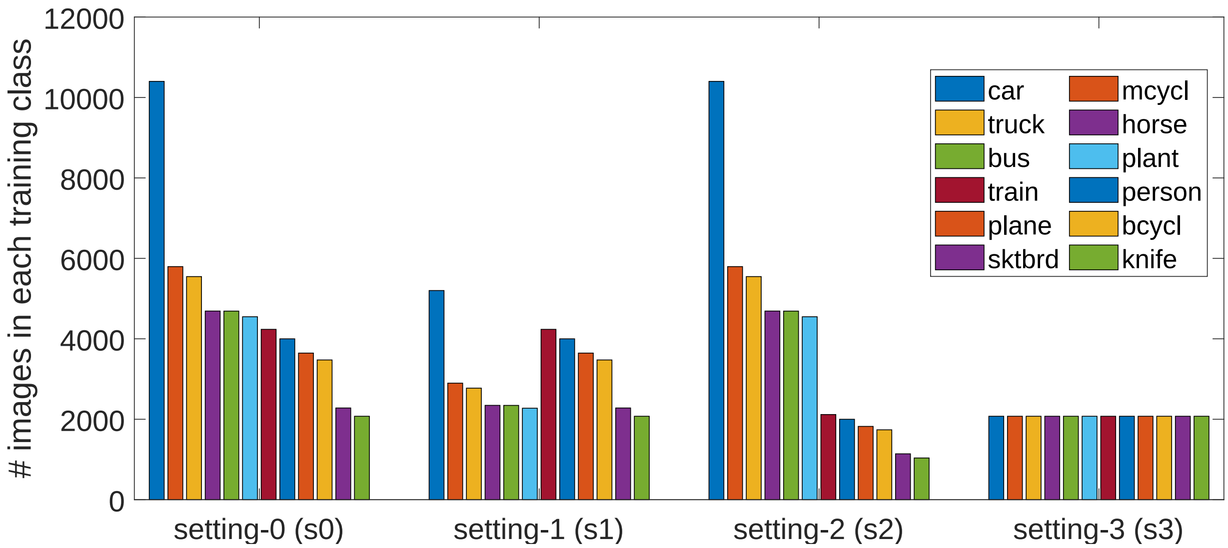


Fig. 1. The number of training samples from each class in different settings on VISDA-C. ‘setting-0 (s0)’ also depicts the label distribution during the testing stage for all the settings.

TABLE II
CLASSIFICATION ACCURACIES (%) OF SHOT-IM ($\beta = 0$) AND SHOT-IM WITH VARYING DIFFERENT DEGREES OF TRAINING CLASS IMBALANCE ON VISDA-C (RESNET-50).

Settings	VISDA-C	car	mcycl	truck	horse	bus	plant	train	person	plane	bcycl	sktbrd	knife	Per-class
	Source-only	53.4	76.9	6.9	44.6	48.9	50.7	89.7	13.1	59.4	10.6	22.1	1.6	39.8
s3	SHOT-IM ($\beta = 0$)	71.7	92.5	0.0	91.7	91.7	90.0	82.3	82.2	93.6	0.0	91.7	2.3	65.8
	SHOT-IM	69.0	89.3	48.9	91.5	79.1	90.2	83.7	81.0	93.3	75.7	77.6	53.7	77.7
s1	SHOT-IM ($\beta = 0$)	68.8	90.6	0.0	86.0	89.9	77.9	80.3	78.5	93.9	0.0	85.5	21.4	64.4
	SHOT-IM	59.5	88.8	52.4	89.8	80.3	87.6	78.1	76.9	93.5	74.8	82.6	30.4	74.6
s0	SHOT-IM ($\beta = 0$)	68.0	87.2	0.0	86.2	89.2	79.7	77.4	69.5	90.2	0.0	0.0	99.4	62.2
	SHOT-IM	53.2	81.4	53.6	89.7	78.2	85.9	86.4	75.8	93.8	80.7	84.8	22.5	73.8
s2	SHOT-IM ($\beta = 0$)	63.8	89.5	0.0	86.2	89.7	81.8	70.8	66.2	86.5	0.0	0.0	99.1	61.1
	SHOT-IM	43.8	71.9	48.2	87.4	70.6	82.6	87.0	78.6	93.2	82.0	76.6	21.6	70.3

the ‘sktbrd’ class after SHOT-IM ($\beta = 0$) becomes 0. When carefully looking at the accuracies of classes ‘truck’ and ‘bcycl’, we find that SHOT-IM ($\beta = 0$) classifies images of ‘bcycl’ to the ‘mcycl’ class and ‘truck’ to the ‘bus’ class, but SHOT-IM achieves higher accuracies for all four classes under different settings. We can conclude that the diversity term within the IM loss is always effective when varying different degrees of class imbalance. Besides, the closer to a uniform vector the ground-truth label distribution of the training target domain is, the better performance the IM loss obtains.

C. INCORPORATION WITH OTHER UDA METHODS

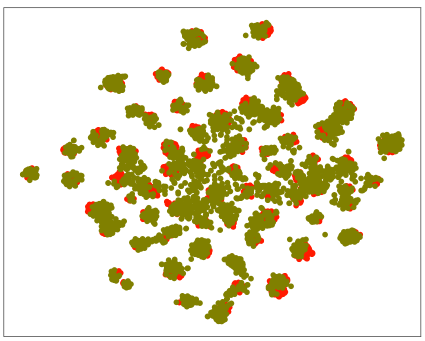
To study the effectiveness of the proposed structure-aware techniques (i.e., self-supervised pseudo-labeling (SPL) and information maximization loss (IM)), we consider two traditional UDA methods (i.e., DANN [1] and CDAN [2]) as baseline methods, and incorporate SPL and IM into them, and report the results on 6 vanilla closed-set UDA tasks in Table III.

As shown in Table III, both SPL and IM greatly improve the adaptation performance of DANN for these UDA tasks. CDAN exploits the semantic information within the adversarial learning strategy, which works much better than DANN. Even though, SPL and IM still boost the performance of CDAN.

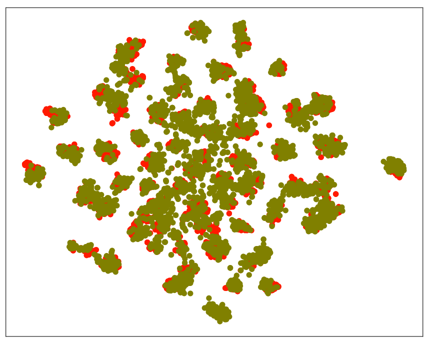
To measure the domain difference, we additionally show the t-SNE feature visualization results of different methods for **Pr** $\rightarrow$ **Re** (closed-set) on Office-Home. In Fig. 2(c-f), it is easy to find that, with the help of the proposed SPL & IM, the target features are well separated and the target features are well aligned with source features, which is in line with the recognition accuracy shown in these sub-captions. Besides, as can be seen from Fig. 2, SHOT can well align the target features with the source features for each class, and features from different classes are clearly separated, which reduces the domain difference better than DANN and CDAN. Different from traditional UDA methods where features from two domains are aligned at the same time, SHOT first learns discriminative source features first, then aligns the target features to the source features, making the optimization objective more clear. Besides, SHOT exploits many target-structure-aware strategies during adaptation, which are also beneficial for domain difference minimization.

TABLE III
CLASSIFICATION ACCURACIES (%) OF TWO CLASSIC DATA-DEPENDENT UDA METHODS INTEGRATED WITH OUR PROPOSED TECHNIQUES ON SIX *vanilla closed-set UDA* TASKS. [SPL: SELF-SUPERVISED PSEUDO-LABELING IN EQ. (7), IM: INFORMATION MAXIMIZATION LOSS IN EQ. (3)]

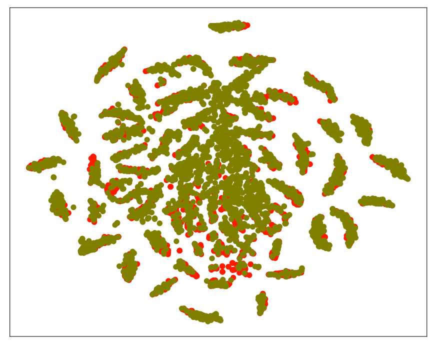
Methods	Office				Office-Home		Avg.	Δ
	A→D	D→A	A→W	W→A	Ar→Cl	Pr→Re		
DANN [1]	78.4	60.8	74.9	62.0	46.6	73.1	66.0	-
w/ SPL	92.9	71.0	91.2	72.2	52.9	80.4	76.8	**+10.8**
w/ IM	90.4	74.6	92.9	75.6	56.5	81.5	78.6	**+12.6**
w/ SPL & IM	94.4	75.4	92.2	73.0	55.8	82.1	78.8	**+12.8**
CDAN+E [2]	92.6	71.8	92.1	70.3	54.4	78.6	76.6	-
w/ SPL	94.5	74.4	91.5	74.9	55.5	80.8	78.6	**+2.0**
w/ IM	92.2	74.7	92.6	74.7	57.5	81.6	78.9	**+2.3**
w/ SPL & IM	95.1	75.0	92.6	75.4	57.1	82.1	79.6	**+3.0**
Source-model-only	80.2	60.3	76.9	63.6	44.5	73.3	66.5	-
SHOT-IM	90.2	72.4	91.1	71.8	55.9	80.9	77.0	**+10.5**
SHOT	93.9	75.3	90.1	75.0	57.7	82.0	79.0	**+12.5**



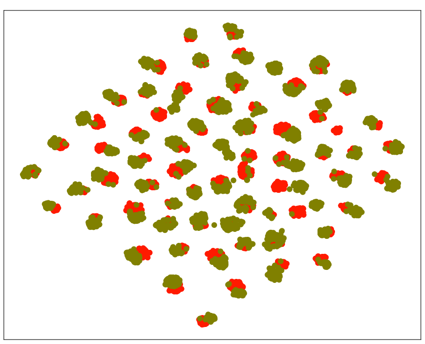
(a) Source-model-only (Acc = 73.31%)



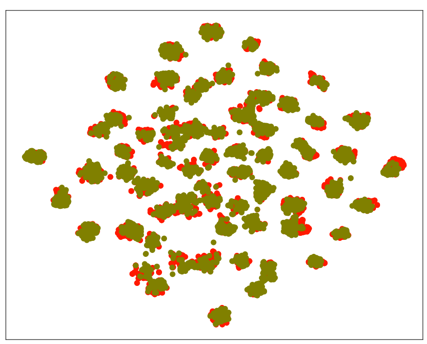
(b) DANN (Acc = 73.38%)



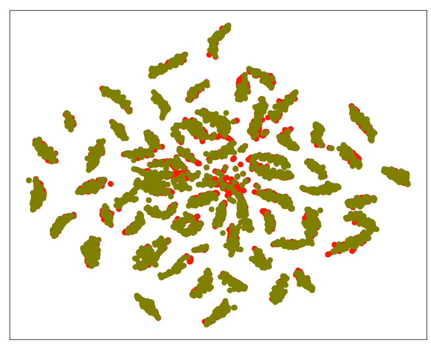
(c) CDAN (Acc = 78.22%)



(d) SHOT (Acc = 82.42%)



(e) DANN + IM + SPL (Acc = 81.71%)



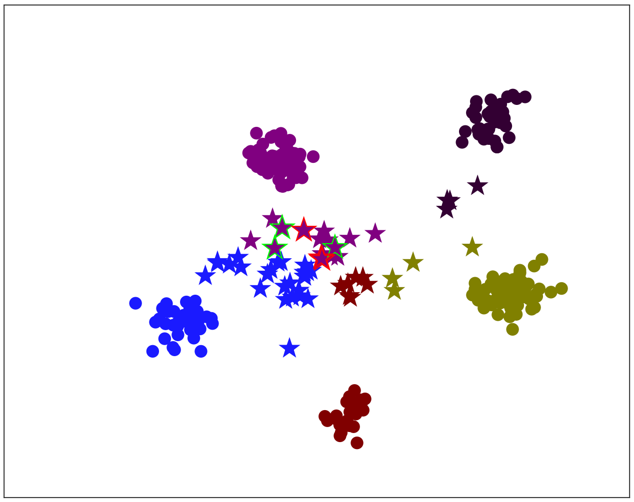
(f) CDAN + IM + SPL (Acc = 81.89%)

Fig. 2. The t-SNE feature visualizations of different methods for **Pr** → **Re** (closed-set) on Office-Home. Circles in red denote source data and circles in olive denote target data.

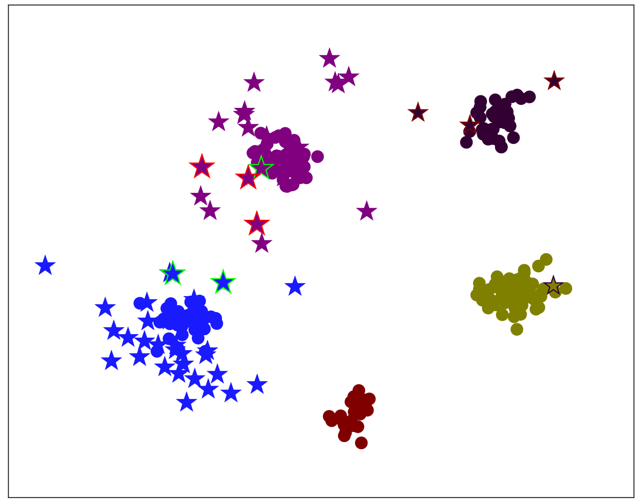
D. HOW DOES THE SELF-SUPERVISED LOSSES WORK

To further explain the contribution of these techniques, we show in Fig. 3 the features learned by different methods. In particular, we select data from the first 5 classes (in alphabetical order) from the source domain and **data from the second class in the target domain** of the closed-set UDA task Ar→Cl on Office-Home and adjust the bottleneck size to 2 for direct feature visualization.

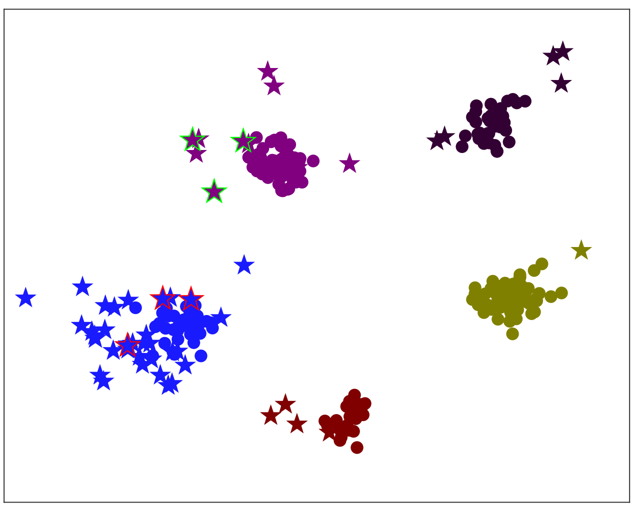
In Fig. 3(a), stars in blue are correctly classified and stars not in blue are mis-classified. Many samples are wrongly classified due to the low-quality representations. Among them, we mainly focus on 6 samples (3 with the green border and 3 with the red border). With the help of IM loss, the target features could be aligned with source features in Fig. 3(b), however, 4 out of 6 samples are still wrongly classified. Incorporated with the self-supervised pseudo-labeling technique, only 3 out of 6 samples are wrongly classified in Fig. 3(c), indicating that the target-specific prototype could provide better pseudo labels to help learn better representations. Taking into consideration the self-supervised rotation prediction objective, all 6 samples could be correctly classified in Fig. 3(d). This may be because self-supervised objective helps focus on semantically meaningful features, which is in line with previous studies [3], [4].



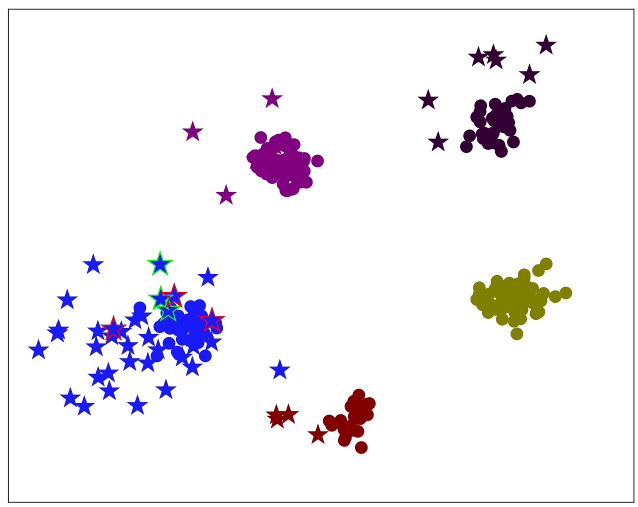
(a) Source-model-only



(b) SHOT-IM



(c) SHOT w/o rotation prediction



(d) SHOT

Fig. 3. Feature visualizations of different methods for a 5-way classification task. Solid circles denote source features and solid stars denote the target features from the second class. Different colors denote different classes. **stars in blue are correctly classified and stars in other colors are wrongly classified.**

E. Different confidence measures within SHOT++

On top of the entropy function (used in SHOT++(ENT)), we have ever tried other choices like the maximum probability (used in SHOT++(MAXP)) and the margin between the largest and the second-largest probability (used in SHOT++(MARP)) to rank the samples. We show the performance for several closed-set UDA tasks with different ranking metrics in Table IV. Besides, SHOT++(RAND) denotes a simple baseline in which we randomly rank the samples.

In terms of the average accuracy in Table IV, SHOT++(MAXP) obtains the best result and SHOT++(ENT) obtains the second-best result. The improvements of SHOT++(RAND) and SHOT++(MARP) over SHOT are quite marginal. Besides, SHOT++(ENT) outperforms SHOT on 6 out of 6 tasks and SHOT++(MAXP) outperforms SHOT on 5 out of 6 tasks. Generally, SHOT++(ENT) and SHOT++(MAXP) achieve similar results, clearly outperforming other counterparts (i.e., SHOT++(MARP) and SHOT++(RAND)).

TABLE IV
CLASSIFICATION ACCURACIES (%) OF SHOT++ WITH DIFFERENT CONFIDENCE METRICS ON SIX *vanilla closed-set UDA* TASKS.

Ranking Metrics	Methods	Office				Office-Home		Avg.	Δ
		A→D	D→A	A→W	W→A	Ar→Cl	Pr→Re		
-	SHOT	93.9	75.3	90.1	75.0	57.7	82.0	79.0	-
$\xi \sim$ rand(0,1)	SHOT++ (RAND)	94.1 ↑	75.4 ↑	90.1	75.2 ↑	57.4	82.4 ↑	79.1	**+0.1**
$p_k - \max_{j \neq k} p_j \ (k = \arg\max_i p_i)$	SHOT++ (MARP)	94.4 ↑	75.5 ↑	90.1	75.2 ↑	57.4	82.4 ↑	79.2	**+0.2**
$p_k \ (k = \arg\max_i p_i)$	SHOT++ (MAXP)	94.5 ↑	76.2 ↑	90.6 ↑	75.5 ↑	57.7	83.0 ↑	79.6	**+0.6**
$\sum_i p_i \log p_i$	SHOT++ (ENT)	94.2 ↑	76.2 ↑	90.3 ↑	75.7 ↑	57.8 ↑	83.0 ↑	79.5	**+0.5**

REFERENCES

[1] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *Proc. ICML*, 2015, pp. 1180–1189.
[2] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *Proc. NeurIPS*, 2018, pp. 1640–1650.
[3] B. Wallace and B. Hariharan, “Extending and analyzing self-supervised learning across domains,” in *Proc. ECCV*, 2020, pp. 717–734.
[4] S. Mishra, K. Saenko, and V. Saligrama, “Surprisingly simple semi-supervised domain adaptation with pretraining and consistency,” *arXiv preprint arXiv:2101.12727*, 2021.