Casting a BAIT for Offline and Online Source-free Domain Adaptation

Shiqi Yang[a,b], Yaxing Wang[c,*], Luis Herranz[a,b], Shangling Jui[d] and Joost van de Weijer[a,b]

[a] *Computer Vision Center, Bellaterra, Spain*

[b] *Department of Computer Science, Universitat Autònoma de Barcelona, Bellaterra, Spain*

[c] *Nankai University, Tianjin, China*

[d] *Huawei Kirin Solution, Shanghai, China*

ABSTRACT

We address the source-free domain adaptation (SFDA) problem, where only the source model is available during adaptation to the target domain. We consider two settings: the offline setting where all target data can be visited multiple times (epochs) to arrive at a prediction for each target sample, and the online setting where the target data needs to be directly classified upon arrival. Inspired by diverse classifier based domain adaptation methods, in this paper we introduce a second classifier, but with another classifier head fixed. When adapting to the target domain, the additional classifier initialized from source classifier is expected to find misclassified features. Next, when updating the feature extractor, those features will be pushed towards the right side of the source decision boundary, thus achieving source-free domain adaptation. Experimental results show that the proposed method achieves competitive results for offline SFDA on several benchmark datasets compared with existing DA and SFDA methods, and our method surpasses by a large margin other SFDA methods under online source-free domain adaptation setting.

1. Introduction

Though achieving great success, typically deep neural networks demand a huge amount of labeled data for training. However, collecting labeled data is often laborious and expensive. It would, therefore, be ideal if the knowledge obtained on label-rich datasets can be transferred to unlabeled data. For example, after training on synthetic images, it would be beneficial to transfer the obtained knowledge to the domain of real-world images. However, deep networks are weak at generalizing to unseen domains, even when the differences are only subtle between the datasets [1]. In real-world situations, a typical factor impairing the model generalization ability is the distribution shift between data from different domains.

Domain Adaptation (DA) [2, 3, 4, 5, 6, 7, 8, 9] aims to reduce the domain shift between labeled and unlabeled target domains. Early works [10] learn domain-invariant features to link the target domain to the source domain. Along with the growing popularity of deep learning, many works benefit from its powerful representation learning ability for domain adaptation. Those methods typically minimize the distribution discrepancy between two domains [11], or deploy adversarial training [12, 13].

However, a crucial requirement in the methodology of these methods is that they require access to the source domain data during the adaptation process to the target domain. Accessibility to the source data of a trained source model is often impossible in many real-world applications, for example deploying domain adaptation algorithms on mobile devices where the computation capacity is limited, or in situations where data-privacy rules limit access to the source domain. Without access to the source domain data, the above methods suffer from inferior performance.

Because of its relevance and practical interest, the *source-free adaptation* (SFDA) setting where the model is first trained on the source domain and has no longer access to the source data afterward, has started to get traction recently [14, 15, 16]. In this paper, we further distinguish between offline and online SFDA. In the offline case, the algorithm can access the target data several times (or epochs) before arriving at a class prediction for each of the samples in the target data. In the online (or streaming) case, the algorithm has to directly predict the label of the incoming target data, meaning that there is only a single pass (or epoch) over the target data. The online scenario is often more realistic, since often an algorithm is expected to directly perform when being exposed to a new domain (as is common in for example robotics applications) and cannot wait with its prediction until it has seen all target data.

Existing method in SFDA have focused on offline SFDA. Among these methods, USFDA [14] addresses universal DA [17] and SF [18] addresses for open-set DA [19]. Both have the drawback of requiring to generate images or features of non-existing categories. SHOT [15] and 3C-GAN [16] address close-set SFDA. 3C-GAN [16] is based on target-style image generation by a conditional GAN, which demands a large computation capacity and is time-consuming. Meanwhile, SHOT proposes to transfer the source hypothesis, i.e. the fixed source classifier, to the target data. Also, the pseudo-labeling strategy is an important step of the SHOT method. However, SHOT has two limitations. First, it needs to access all target data to compute the pseudo labels, only after this phase it can start adaptation to the target domain. This is infeasible for online streaming applications where the system is expected to directly process the target data and data cannot be revisited. Secondly, it

[*] Corresponding author: yaxing@nankai.edu
ORCID(s):

heavily depends on pseudo-labels being correct. Therefore, some wrong pseudo-labels may compromise the adaptation process.

Our method is inspired by the diverse classifiers based DA method MCD [20]. However, that work fails for SFDA. Like that work we also deploy two classifiers to align target with source classifier. In our method, after getting the source model, we propose to freeze the classifier head of the source model during the whole adaptation process. The decision boundary of this source classifier serves as an anchor for SFDA. Next, we add an extra classifier (called *bait* classifier) initialized from the source classifier (referred to as *anchor* classifier). The bait classifier is expected to find those target features that are misclassified by the source classifier. By encouraging the two classifiers to have similar predictions, the feature extractor will push target features to the correct side of the source decision boundary, thus achieving adaptation. In the experiments, we show that our method, dubbed BAIT, achieves competitive results compared with methods using source data and also other SFDA methods. Moreover, other than SHOT, our method can directly start adaptation to the target domain when target data arrives, and does not require a full pass through the target data before starting adaptation. As a consequence, our method obtains superior results in the more realistic setting of online source-free domain adaptation.

We summarize our contributions as follows:

- We propose a new method for the challenging source-free domain adaptation scenario. under either online or offline setting. Our method does neither require image generation as in [16, 14, 18] and does not require the usage of pseudo-labeling [15].

- Our method prevents the need for source data by deploying an additional classifier to align target features with the source classifier. We thus show that the previously popular diverse classifiers methods designed for DA ([20]) can be extended to SFDA by introducing a fixed classifier, entropy based splitting and a class-balance loss.

- We demonstrate that the proposed BAIT approach obtains similar results or outperforms existing DA and SFDA methods on several datasets.

2. Related Works

Domain adaptation with source data. Early moment matching methods align feature distributions by minimizing the feature distribution discrepancy, including methods such as DAN [2] which deploys Maximum Mean Discrepancy. CORAL [21] matches the second-order statistics of the source to target domain. Inspired by adversarial learning, CDAN [22] trains a deep networks conditioned on several sources of information. DIRT [23] performs domain adversarial training with an added term that penalizes violations of the cluster assumption. Domain adaptation has also been tackled from other perspectives. DAMN [24] introduces a framework where each domain undergoes a different sequence of operations.

Domain adaptation without source data. All these methods, however, require access to source data during adaptation. Recently, USFDA [14] and FS [18] explore the source-free setting, but they focus on the universal DA task [17] and open-set DA [19], where the label spaces of source and target domain are not identical. And their proposed methods are based on generation of simulated negative labeled samples during source straining period, in order to increase the generalization ability for unknown class. Most relevant works are SHOT [15] and 3C-GAN [16], both are for close-set DA. SHOT needs to compute and update pseudo labels before updating model, which has to access all target data and may also have negative impact on training from the noisy pseudo labels, and 3C-GAN needs to generate target-style training images based on conditional GAN, which demands large computation capacity. Instead of synthesizing target images or using pseudo labels, our method introduces an additional classifier to achieve feature alignment with the fixed source classifier. And recently there is also research line addressing SFDA by clustering [25, 26, 27], however, those methods need to build memory bank for neighbors retrieving, thus they are not feasible for online domain adaptation. Our work is inspired by MCD [20], however, it is more efficient and performs well under the source-free setting. It is important to note that for MCD source supervision is crucial during adaptation on target.

3. BAIT for Source-Free Domain Adaptation

We start by introducing our normal offline source-free domain adaptation method. Finally, we will extend this method to the online case where target data are only seen once.

We denote the labeled source domain data with n_s samples as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where y_i^s is the corresponding label of x_i^s, and the unlabeled target domain data with n_t samples as $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$, and the number of classes is K. Usually, DA methods eliminate the domain shift by aligning the feature distribution between the source and target domains. Unlike the normal setting, we consider the more challenging SFDA setting which during adaptation to the target data has no longer access to the source data, and has only access to the model trained on the source data.

3.1. Source classifier as anchor

We decompose the neural network into two parts: a feature extractor f, and a classifier head C_1 which only contains one fully connected layer (with weight normalization). We first train the baseline model on the labeled source data $\mathcal{D}_s$ with standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{k=1}^{K} I_{[k=y_i^s]} \log p_k(x_i^s) \tag{1}$$

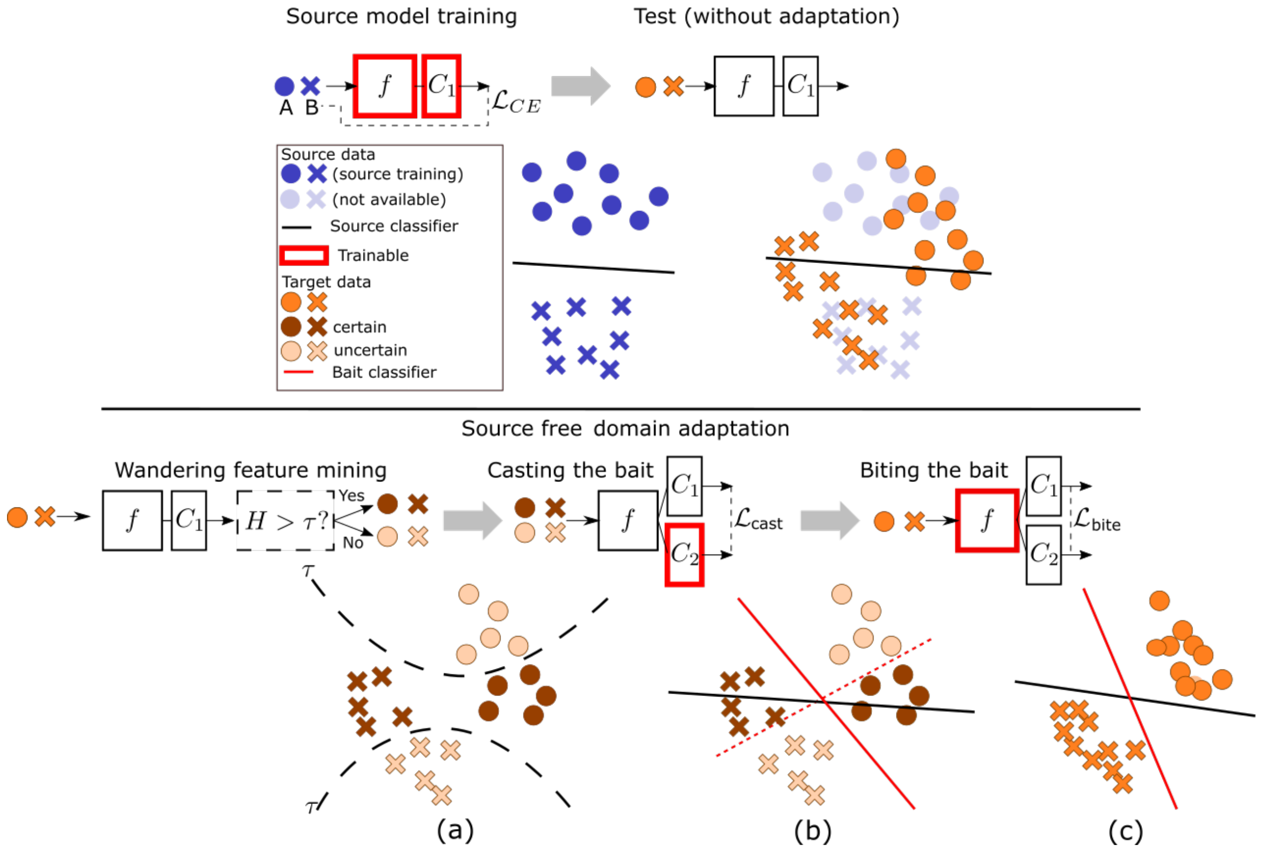


Figure 1: Illustration of training process. The top shows that the source-training model fails on target domain due to domain shift. The bottom illustrates our adaptation process. (a): splitting feature in **current batch** into 2 groups by the prediction entropy H and the threshold τ, (b) then increasing prediction divergence between two classifiers for uncertain features but keep the prediction unchanged for uncertain features, meanwhile maximizing KL divergence can also prevent bait classifier from moving to the undesirable position (dashed red line). (c): training the feature extractor pushes all features to the same side of C_1 and C_2.

where the p_k is the k-th element of the softmax output, and $I_{[z]}$ is the indicator function which is 1 if z is true, and 0 otherwise.

A closer look at the training process of DA methods unveils that the feature extractor aims to learn a discriminative representation, and the classifier strives to distinguish the representations of the various classes. DA methods tackle domain shift by aligning the feature distribution (from the feature extractor) of the source and target domains. A successful alignment of the features means that the features produced by the feature extractor f from both domains will be classified correctly by the classifier head.

As shown in Fig 1(left), due to the domain shift, the cluster of target features generated by the source-training feature extractor will deviate from the source class prototype, meaning some target features will locate at the wrong side of the source decision boundary. Similar to [14, 15], we freeze the source-trained classifier C_1. This implicitly allows us to store the relevant information from the source domain, *i.e.*, the position of the source decision boundary. With the source classifier as an anchor in the feature space, we hope to push the target features towards the right side of the decision boundary. Hereafter, we refer to classifier C_1 as the *anchor classifier*.

3.2. Second classifier as bait

For the fixed anchor classifier to be successful for source-free domain adaptation, we require to address two problems. First, part of the target data will not be well classified (have uncertain predictions) due to the domain shift, and this data needs to be identified. Secondly, we have to adapt the feature extractor in such a way that this data can subsequently be classified correctly by the anchor classifier. Therefore, we propose the BAIT method that is a two-step algorithm which exactly addresses these two problems. Our method is shown in Fig. 1(right). After training the model on the source data, we get a feature extractor f, and an anchor classifier C_1. We fix C_1 in the subsequent training periods, and use it to initialize an extra classifier C_2. The extra classifier C_2 is optimised to find target features which are not clearly classified by C_1. Next, those target features are to be pulled towards the right side of source classifier. Hereafter, we refer to the classifier C_2 as the *bait classifier*.

In order to train the desired C_2, we propose a 2-step training policy which alternates between training the bait classifier C_2 and feature extractor f. It is inspired by diverse classifiers based DA method, such as MCD [20]. Unlike those methods which train all classifiers along with source data, our method addresses source-free domain adaptation with the fixed anchor classifier and the learnable bait classifier. We experimentally show that the original MCD cannot handle SFDA while our proposed method performs well under this setting.

Step 1: casting the bait. In step 1, we only train bait classifier C_2, and freeze feature extractor f. As shown in Fig. 1, due to the domain shift, some target features will not locate on the right side of the source decision boundary, which is also referred to as misalignment [4, 28]. In order to align target features with the source classifier, we use the bait classifier to find the those features at the wrong side of the anchor classifier/decision boundary (uncertain features).

Therefore, before adaptation, we split the features of the current **mini-batch** of data into two sets: the uncertain $\mathcal{U}$ and certain set $\mathcal{C}$, as shown in Fig. 1 (a), according to their prediction entropy:

$$\begin{aligned} \mathcal{U} &= \left\{x|x \in \mathcal{D}_t, H\left(p^{(1)}(x)\right) > \tau\right\}, \\ \mathcal{C} &= \left\{x|x \in \mathcal{D}_t, H\left(p^{(1)}(x)\right) \leq \tau\right\} \end{aligned} \tag{2}$$

where $p^{(1)}(x) = \sigma(C_1(f(x))$ is the prediction of the anchor classifier (σ represents the softmax operation) and $H(p(x)) = -\sum_{i=1}^{K} p_i \log p_i$. The threshold τ is estimated as a percentile of the entropy of $p_1(x)$ in the mini-batch. We empirically found that choosing τ such that the data is equally split between the certain and uncertain set provided good results (also see ablation).

Having identified the certain and uncertain features, we now optimize the bait classifier to reduce the symmetric KL divergence for the certain features, while increasing it for the uncertain features. As a consequence, the two classifiers will agree for the certain features but disagree for the uncertain features. This is achieved by following objective:

$$\begin{aligned} \mathcal{L}_{\text{cast}}(C_2) = &\sum_{x \in \mathcal{C}} D_{SKL}(p^{(1)}(x), p^{(2)}(x)) \\ &- \sum_{x \in \mathcal{U}} D_{SKL}(p^{(1)}(x), p^{(2)}(x)) \end{aligned} \tag{3}$$

where D_{SKL} is the symmetric KL divergence: $D_{SKL}(a, b) = \frac{1}{2}\left(D_{KL}(a|b) + D_{KL}(b|a)\right)$. Note that $KL(p^{(2)}||p^{(1)}) = -H(p^{(2)}) - \sum p^{(2)} log p^{(1)}$. Instead of using L1 distance like MCD [20], the advantage of maximizing KL divergence is that it can prevent the bait classifier from moving to the undesirable place, as the dashed red line shown in the Fig. 1(b), since minimizing entropy will encourage the decision boundary not to go across the dense feature region according to cluster assumption [29, 30, 23].

As shown in Fig. 1 (a-b), given that C_2 is initialized from C_1, increasing the KL-divergence on the uncertain set between two classifiers will drive the boundary of C_2 to those features with higher entropy. Decreasing it on the certain set encourages the two classifiers to have similar predictions for those features. This will ensure that the features with lower entropy (of high possibility with correct prediction) will stay on the same side of the classifier.

Step 2: biting the bait. In this stage, we only train the feature extractor f, aiming to pull the target features towards the same side of two classifiers.Specifically, we update the feature extractor by minimizing the proposed bite loss:

$$\mathcal{L}_{\text{bite}}(f) = \sum_{i=1}^{n_t} \sum_{k=1}^{K} [-p_{i,k}^{(2)} \log p_{i,k}^{(1)} - p_{i,k}^{(1)} \log p_{i,k}^{(2)}] \tag{4}$$

By minimizing this loss, the prediction distribution of the bait classifier should be similar to that of the anchor classifier and vice verse, which means target features are excepted to locate on the same sides of the two classifiers.

Intuitively, as shown in Fig. 1 (c), minimizing the bite loss $\mathcal{L}_{\text{bite}}$ will push target features towards the right direction of the decision boundary. Metaphorically, in this stage the anchor classifier bites the bait (those features with different predictions from anchor and bait classifier) and pushes it towards the anchor classifier.

Additionally, in order to avoid the degenerate solutions, which allocate all uncertain features to some specific class, we adopt the class-balance loss $\mathcal{L}_b$ to regularize the feature extractor [31, 32]:

$$\mathcal{L}_{\text{b}}(f) = \sum_{k=1}^{K} [KL(\bar{p}_k^{(1)}(x))||q_k) + KL(\bar{p}_k^{(2)}(x))||q_k)] \tag{5}$$

where $\bar{p}_k = \frac{1}{n_t} \sum_{x \in \mathcal{D}_t} p_k(x)$ is the empirical label distribution, and q is uniform distribution $q_k = \frac{1}{K}, \sum_{k=1}^{K} q_k = 1$. With the class-balance loss $\mathcal{L}_b$, the model is expected to have a more balanced prediction.

Online source-free domain adaptation As discussed in the introduction, for many applications the current paradigm of offline SFDA is not realistic. This paradigm requires the algorithm to first collect all target data (and be able to process it multiple times) before arriving at a class prediction for each target dataset sample. In the online case, the algorithm has to directly provide class predictions as the target data starts arriving. This scenario is for example typical in robotics applications, where the robot has to directly function when arriving in a new environment.

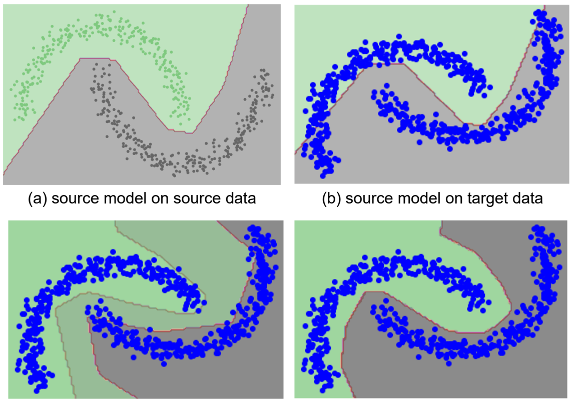
Our proposed method can be straightforwardly extended to online SFDA. Since in this case the predictions of the fixed classifier have only a low reliability, we found that it was beneficial in the online setting to remove the entropy based splitting. During the adaptation, the target data are only accessible once, *i.e.*, we only train one epoch.

4. Experiments

In the following, we first test our method on a toy dataset. Then we provide detailed experiments under offline setting. Finally, we evaluate our method under online setting.

4.1. Experiment on Twinning moon dataset

We carry out our experiment on the twinning moon dataset. For this data set, the data from the source domain are


(a) source model on source data

(b) source model on target data

(c) after stage 1 adaptation

(d) after stage 2 adaptation

Figure 2: Toy experiment on the twinning moon 2D dataset. The blue points refer to target data. The green and gray refer to source data. Decision boundaries after training model only on the source data and testing on source (a) and target (b) data. (c) After stage 1 training in the middle of adaptation with only target data. The two borderlines denote two decision boundaries (with C_1 in red). (d) After stage 2 training, the two decision boundaries almost coincide.

Table 1
Results for various domain adaptations on Office-31 and VisDA. A2D refers to an adaptation from domain A(Amazon) to D(DSLR), etc. The three methods at the bottom are **source-free** methods. 'Avg' means average precision over all tasks, while 'Per-class' means average per-class accuracy over 12 classes.

Method	**Office-31**							**VisDA**
	A2D	A2W	D2W	W2D	D2A	W2A	Avg	Per-class
Source model	74.1	95.3	99.0	80.1	54.0	56.3	76.5	46.3
CDAN [22]	93.1	98.2	100	89.8	70.1	68.0	86.6	70.0
MDD [33]	94.5	98.4	100	93.5	74.6	72.2	74.6	
DMRL [34]	93.4	90.8	99.0	100.0	73.0	71.2	87.9	75.5
BNM [35]	90.3	91.5	98.5	100.0	70.9	71.6	87.1	-
SRDC [36]	95.8	95.7	99.2	100.0	76.7	77.1	**90.8**	-
MCC [37]	95.6	95.4	98.6	100.0	72.6	73.9	89.4	78.8
DWL [38]	91.2	89.2	99.2	100.0	73.1	69.8	87.1	77.1
3C-GAN [16]	92.7	93.7	98.5	99.8	75.3	77.8	89.6	81.6
SHOT [15]	93.1	90.9	98.8	99.9	74.5	74.8	88.7	82.9
Ours	92.0	94.6	98.1	100.0	74.6	75.2	89.1	**83.0**

represented by two inter-twinning moons, which contain 300 samples each. We generate the data in the target domain by rotating the source data by 30°. Here the rotation degree can be regarded as the domain shift. First we train the model only on the source domain, and test the model on all domains. As shown in Fig. 2(a) and (b), due to the domain shift the model performs worse on the target data. Then we adapt the model to the target domain with the anchor and bait classifiers, without access to any source data. As shown in Fig 2(c) during adaptation the bait loss moves the decision boundary[1] of the bait classifier to different regions than the anchor classifier. After adaptation the two decision boundaries almost coincide and both classifiers give the correct prediction, as shown in Fig. 2(d).

Table 2
Results on domain adaptation on Office-Home. The two methods at the bottom are **source-free** methods. 'Avg' means average precision over all tasks.

Method	**Office-Home**												
	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg
Source model	37.0	62.2	70.7	46.6	55.1	60.3	46.1	32.0	68.7	61.8	39.2	75.4	54.6
CDAN [22]	49.0	69.3	74.5	54.4	66	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
MDD [33]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
BNM [35]	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
SRDC [36]	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
SHOT [15]	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**
Ours	57.4	77.5	82.4	68.0	77.2	75.1	67.1	55.5	81.9	73.9	59.5	84.2	71.6

4.2. Offline Source-free Domain Adaptation

Datasets. We use three benchmark datasets. **Office-31** [39] contains 3 domains (Amazon denoted as A, Webcam denoted as W, DSLR denoted as D) with 31 classes and 4,652 images. **Office-Home** [40] contains 4 domains (Real denoted as R, Clipart denoted as C, Art denoted as A, Product denoted as P) with 65 classes and a total of 15,500 images. **VisDA-2017** [41] (denoted as VisDA) is a more challenging dataset, with 12-class synthesis-to-real object recognition tasks, its source domain contains 152k synthetic images while the target domain has 55k real object images.

Model details We adopt the backbone of ResNet-50 [42] (for office datasets) or ResNet-101 (for VisDA) along with an extra fully connected (fc) layer as feature extractor, and a fc layer as classifier head. We adopt SGD with momentum 0.9 and batch size of 128 on all datasets. On the source domain, the learning of the ImageNet pretrained backbone and the newly added layers are 1e-3 and 1e-2 respectively, except for the ones on VisDA, which are 1e-4 and 1e-3 respectively. We further reduce the learning rate 10 times training on the target domain. We train 20 epochs on the source domain, and 30 epochs on the target domain. All experiments are conducted on a single RTX 6000 GPU. All results are reported from the classifier C_1, and are the average across three running with random seeds.

Quantitative Results. The results under offline setting on the three datasets are shown in Tab. 1 and Tab. 2. In these tables, the top part shows results for the normal setting with access to source data during adaptation. The bottom one shows results for the source-free setting. As reported in Tab. 1 and Tab. 2, our method outperforms most methods which have access to source data on all these datasets.

The proposed method still obtains the comparative performance when comparing with current source-free methods. In particular, our method surpasses SHOT [15] by 0.1%, and 3C-GAN [16] by 1.4% on the more challenging VisDA dataset (Tab. 1), and gets closer results on Office-Home (Tab. 2) compared to SHOT. Note that 3C-GAN highly relies on the extra synthesized data. On Office-31 (Tab. 1), the proposed BAIT achieves better result than SHOT, and competitive result to 3C-GAN. The reported results clearly demonstrate the efficacy of the proposed method without access to the source data during adaptation.

Ablation Study. We conduct a detailed ablation study to isolate the performance gain due to key components of our

[1] Note here the decision boundary is from the whole model, since the input are data instead of features.

Table 3
(**Left**) Ablation study. fix means fixing the first classifier, spl means entropy splitting, $\mathcal{L}_b$ is the class-balance loss. (**Right**) Ablation study on τ. All uncertain means no splitting actually.

SF	fix	spl	$\mathcal{L}_b$	Avg.
×	×	×	×	65.3
√	×	×	×	60.8
√	√	×	×	67.3
√	×	√	×	60.2
√	×	×	√	52.7
√	√	√	×	68.2
√	√	√	√	**71.6**

Office-Home	Avg.
BAIT (τ as all uncertain)	70.8
BAIT (τ as 75% uncertain)	71.2
BAIT (τ as 50% uncertain, in paper)	**71.6**
BAIT (τ as 25% uncertain)	70.2

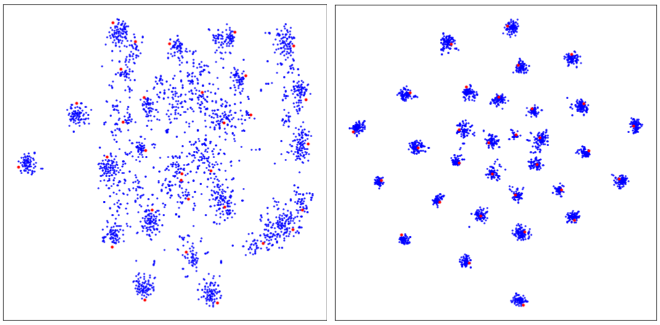


Figure 3: t-SNE visualization (Left: Source only; Right: BAIT) for features from the target domain on W− >A (Office-31). The red points are the class prototype from C_1. Best seen in screen.

method. Note the existing domain adaptation datasets do not provide train/validation/test splitting, here we directly conduct the ablation study on test set, just as all existing methods did. We start from a variant of MCD which is reproduced by ourselves as a baseline (the first and second row in Tab. 3), note we replace the L1 distance in the original MCD with Eq.2 and Eq.3 used in our paper. As shown by the results in Tab. 3 on the Office-Home, if removing the access to source data (SF), significant degrading will occur for MCD. Then with our proposed modules on top of this baseline: fixing the first classifier (fix), entropy splitting (spl) and class-balance loss ($\mathcal{L}_b$), it performs well under SFDA setting. The experimental results show the effectiveness of the proposed method and the importance of all components. In addition, we ablate $\mathcal{L}_{cast}$ which is used to train the auxiliary classifier, and $\mathcal{L}_{bite}$ which trains the feature extractor. Both are necessary components of our method, removing any one of these results in very bad performance: removing L_{cast} obtains 45% and removing $\mathcal{L}_{bite}$ gets only 8%.

We also report results with different τ. In all experiments, we have set τ as to select half of the *current batch* as certain and uncertain set. Here, in Tab. 3, we also choose τ to select 100%, 75% or 25% of the *current batch* as the uncertain set, the results show our method is not very sensitive to the choice of this hyperparameter, we posit that this is because of the random mini-batch sampling where the same image can be grouped into both certain and uncertain set in different batches during training.

Embedding visualization. Fig. 3 shows the t-SNE visualization of target features obtained with the source model and after adaptation with BAIT. Target features form more compact and clear clusters after BAIT than in the source model, indicating that BAIT produces more discriminative features. We also show the class prototypes (red points) which are the weights of classifier C_1, it shows target features cluster around the corresponding prototypes.

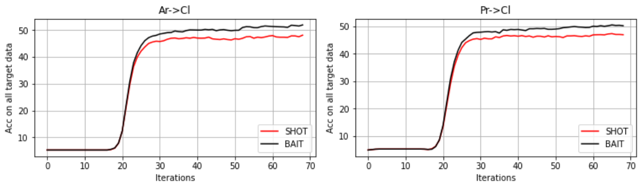


Figure 4: Accuracy on the whole target data during online adaptation on Office-Home.

4.3. Online Source-free Domain Adaptation

We also report results for the online setting, where all target data can only be accessed once, *i.e.*, training for only one epoch. All datasets and model details stay the same as in the offline setting in Sec. 4.2. After one epoch training, we evaluate the model on the target data. This setting is important for some online streaming situations, where the system is expected to directly process the target data and data cannot be revisited. Note under this setting we abandon the entropy splitting. We reproduce SHOT [15] under this setting as the authors released their code. As shown in Tables 4, our BAIT outperforms SHOT on all three datasets. The accuracy curve for two tasks of Office-Home during adaptation is shown in Fig. 4. They show our method is outperforms SHOT. The reason of lower starting accuracy is because we initialize all BN layers with zero means and variance equal to one before adaptation, which was found to lead to better results. Note here the model cannot access all target data in each mini-batch training, thus SHOT can only use the current mini-batch to compute pseudo labels. This means that the computed pseudo labels are quite similar with the naive pseudo label from the model, thereby compromising the performance. This is the reason SHOT gets lower results than BAIT. BAIT is an extension of MCD. Tables 4 shows that indeed the proposed changes do considerably impact performance and our method without source data even outperforms MCD with source data.

5. Conclusion

There are many practical scenarios where source data may not be available (e.g. due to privacy or availability restrictions) or may be expensive to process. In this paper we study this challenging yet promising domain adaptation setting (i.e. SFDA), and propose BAIT, a fast and effective approach. BAIT aligns target features with fixed source classifier via an extra bait classifier that locates uncertain target features and drags them towards the right side of the source decision boundary. The experimental results show that BAIT achieves competitive performance on several datasets

Table 4
(**Top**) Results on **online** source-free domain adaptation on Office-31 and Office-Home. ’Avg’ means average precision over all tasks. The three methods at the bottom are **source-free** methods. (**Bottom**) Results on **online** source-free domain adaptation on VisDA. ’Per-class’ means average per-class accuracy over 12 classes. The three methods at the bottom are **source-free** methods.

Method	**Office-31**							**Office-Home**												
	A2D	A2W	D2W	W2D	D2A	W2A	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg
MCD (*w/* source) [20]	87.9	86.1	92.3	96.9	62.2	65.3	81.8	46.3	65.3	74.9	57.2	64.3	66.0	55.1	45.3	75.1	66.7	48.4	78.1	61.9
MCD (*w/o* source) [20]	85.8	82.4	91.0	96.0	59.6	62.3	79.5	44.2	63.7	72.7	50.6	63.9	60.2	54.8	42.4	73.0	60.2	47.5	76.9	59.2
SHOT [15]	87.8	85.2	96.0	99.6	70.1	68.3	84.6	48.1	72.0	76.2	59.0	68.9	67.8	58.7	47.0	77.3	70.0	53.8	80.1	64.9
BAIT (ours)	90.8	86.2	96.5	99.8	71.5	69.6	**85.7**	51.0	72.9	77.4	60.9	71.0	68.7	60.7	49.3	78.2	70.2	54.5	80.4	**66.3**

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
MCD (*w/* source)	77.8	53.1	79.6	55.3	80.1	64.3	78.9	62.6	68.7	40.5	77.2	20.4	63.2
MCD (*w/o* source)	71.4	47.7	74.7	50.2	76.8	62.3	73.1	60.7	65.4	37.5	73.2	17.6	59.2
SHOT	89.5	59.9	85.6	57.3	87.6	72.4	89.3	70.0	75.1	45.3	82.4	39.8	71.2
BAIT (ours)	93.2	66.2	87.1	59.1	90.2	76.9	92.1	83.4	80.6	49.5	87.7	45.7	**76.0**

under the offline setting, and surpasses other SFDA methods in the more realistic online setting.

Acknowledgement We acknowledge the support from Huawei Kirin Solution, and the project PID2019-104174GB-I00 (MINECO, Spain), TED2021-132513B-I00, and PID2021-128178OB-I00 (MICINN, Spain), and the Ramón y Cajal fellowship RYC2019-027020-I, and the CERCA Programme of Generalitat de Catalunya.

References

[1] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: CVPR, 2014, pp. 1717–1724.

[2] M. Long, Y. Cao, J. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, ICML.

[3] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: CVPR, 2017, pp. 7167–7176.

[4] S. Cicek, S. Soatto, Unsupervised domain adaptation via regularized conditional alignment, in: ICCV, 2019, pp. 1416–1425.

[5] A. S. Mozafari, M. Jamzad, Cluster-based adaptive svm: A latent subdomains discovery method for domain adaptation problems, Computer Vision and Image Understanding 162 (2017) 116–134.

[6] M. Wang, P. Li, L. Shen, Y. Wang, S. Wang, W. Wang, X. Zhang, J. Chen, Z. Luo, Informative pairs mining based adaptive metric learning for adversarial domain adaptation, Neural Networks 151 (2022) 238–249.

[7] M. Wang, W. Wang, B. Li, X. Zhang, L. Lan, H. Tan, T. Liang, W. Yu, Z. Luo, Interbn: Channel fusion for adversarial unsupervised domain adaptation, in: ACM MM, 2021, pp. 3691–3700.

[8] M. Wang, S. Wang, Y. Wang, W. Wang, T. Liang, J. Chen, Z. Luo, Boosting unsupervised domain adaptation: A fourier approach, Knowledge-Based Systems (2023) 110325.

[9] M. Wang, S. Wang, W. Wang, L. Shen, X. Zhang, L. Lan, Z. Luo, Reducing bi-level feature redundancy for unsupervised domain adaptation, Pattern Recognition (2023) 109319.

[10] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2066–2073.

[11] M. Long, Y. Cao, Z. Cao, J. Wang, M. I. Jordan, Transferable representation learning with deep adaptation networks, IEEE TPAMI 41 (12) (2018) 3071–3085.

[12] Y. Zhang, H. Tang, K. Jia, M. Tan, Domain-symmetric networks for adversarial domain adaptation, in: CVPR, 2019, pp. 5031–5040.

[13] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, T. Xiang, Stochastic classifiers for unsupervised domain adaptation, in: CVPR, 2020, pp. 9111–9120.

[14] J. N. Kundu, N. Venkat, R. V. Babu, Universal source-free domain adaptation, CVPR.

[15] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, ICML.

[16] R. Li, Q. Jiao, W. Cao, H.-S. Wong, S. Wu, Model adaptation: Unsupervised domain adaptation without source data, in: CVPR, 2020, pp. 9641–9650.

[17] K. You, M. Long, Z. Cao, J. Wang, M. I. Jordan, Universal domain adaptation, in: CVPR, 2019, pp. 2720–2729.

[18] J. N. Kundu, N. Venkat, A. Revanur, R. V. Babu, et al., Towards inheritable models for open-set domain adaptation, in: CVPR, 2020, pp. 12376–12385.

[19] K. Saito, S. Yamamoto, Y. Ushiku, T. Harada, Open set domain adaptation by backpropagation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 153–168.

[20] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: CVPR, 2018, pp. 3723–3732.

[21] B. Sun, J. Feng, K. Saenko, Return of frustratingly easy domain adaptation, in: AAAI, 2016.

[22] M. Long, Z. Cao, J. Wang, M. I. Jordan, Conditional adversarial domain adaptation, in: NeurIPS, 2018, pp. 1647–1657.

[23] R. Shu, H. H. Bui, H. Narui, S. Ermon, A dirt-t approach to unsupervised domain adaptation, ICLR.

[24] R. Bermudez Chacon, M. Salzmann, P. Fua, Domain-adaptive multi-branch networks, in: 8th ICLR, 2020.

[25] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, S. Jui, Generalized source-free domain adaptation, in: ICCV, 2021.

[26] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, S. Jui, Exploiting the intrinsic neighborhood structure for source-free domain adaptation, in: NeurIPS, 2021.

[27] S. Yang, Y. Wang, K. Wang, S. Jui, J. van de Weijer, Attracting and dispersing: A simple approach for source-free domain adaptation, in: NeurIPS, 2022.

[28] X. Jiang, Q. Lao, S. Matwin, M. Havaei, Implicit class-conditioned domain alignment for unsupervised domain adaptation, arXiv preprint arXiv:2006.04996.

[29] O. Chapelle, A. Zien, Semi-supervised classification by low density separation., in: AISTATS, Vol. 2005, Citeseer, 2005, pp. 57–64.

[30] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, in: NeurIPS, 2005, pp. 529–536.

[31] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: ICCV, 2017, pp. 5736–5745.

[32] Y. Shi, F. Sha, Information-theoretical learning of discriminative clusters for unsupervised domain adaptation, arXiv preprint arXiv:1206.6438.

[33] Y. Zhang, T. Liu, M. Long, M. Jordan, Bridging theory and algorithm for domain adaptation, in: International Conference on Machine Learning, 2019, pp. 7404–7413.

[34] Y. Wu, D. Inkpen, A. El-Roby, Dual mixup regularized learning for adversarial domain adaptation, ECCV.

[35] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, Q. Tian, Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations, CVPR.

[36] H. Tang, K. Chen, K. Jia, Unsupervised domain adaptation via structurally regularized deep clustering, in: CVPR, 2020, pp. 8725–8735.

[37] Y. Jin, X. Wang, M. Long, J. Wang, Minimum class confusion for versatile domain adaptation, ECCV.

[38] N. Xiao, L. Zhang, Dynamic weighted learning for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 15242–15251.

[39] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: European conference on computer vision, Springer, 2010, pp. 213–226.

[40] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: CVPR, 2017, pp. 5018–5027.

[41] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, Visda: The visual domain adaptation challenge, arXiv preprint arXiv:1710.06924.

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.