Generalized Source-free Domain Adaptation

Shiqi Yang[1], Yaxing Wang[1,2]*, Joost van de Weijer[1], Luis Herranz[1], Shangling Jui[3]
[1] Computer Vision Center, Universitat Autonoma de Barcelona, Barcelona, Spain
[2] PCALab, Nanjing University of Science and Technology, China
[3] Huawei Kirin Solution, Shanghai, China
{syang,yaxing,joost,lherranz}@cvc.uab.es, jui.shangling@huawei.com

Abstract

Domain adaptation (DA) aims to transfer the knowledge learned from a source domain to an unlabeled target domain. Some recent works tackle source-free domain adaptation (SFDA) where only a source pre-trained model is available for adaptation to the target domain. However, those methods do not consider keeping source performance which is of high practical value in real world applications. In this paper, we propose a new domain adaptation paradigm called Generalized Source-free Domain Adaptation (G-SFDA), where the learned model needs to perform well on both the target and source domains, with only access to current unlabeled target data during adaptation. First, we propose local structure clustering (LSC), aiming to cluster the target features with its semantically similar neighbors, which successfully adapts the model to the target domain in the absence of source data. Second, we propose sparse domain attention (SDA), it produces a binary domain specific attention to activate different feature channels for different domains, meanwhile the domain attention will be utilized to regularize the gradient during adaptation to keep source information. In the experiments, for target performance our method is on par with or better than existing DA and SFDA methods, specifically it achieves state-of-the-art performance (85.4%) on VisDA, and our method works well for all domains after adapting to single or multiple target domains. Code is available in https://github.com/Albert0147/G-SFDA.

1. Introduction

Though achieving great success, deep neural networks typically require a large amount of labeled data for training. However, collecting labeled data is often laborious and expensive. To tackle this problem, *Domain Adaptation* (DA) methods aim to transfer knowledge learned from label-rich datasets (source domains) to other unlabeled datasets (target domains), by reducing the domain shift between labeled source and unlabeled target domains.

A crucial requirement in most DA methods is that they require access to the source data during adaptation, which is often impossible in many real-world applications, such as 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. Because of its relevance and practical interest, the *source-free domain adaptation* (SFDA) setting, where instead of source data only source pretrained model is available, has started to get traction recently [14, 15, 18, 20, 46]. Among these methods, SHOT [20] and 3C-GAN [18] are most related to this paper which is for close-set DA where source and target domains have the same categories. 3C-GAN [18] is based on target-style image generation by a conditional GAN, and SHOT [20] proposes to transfer the source hypothesis, i.e. the fixed source classifier, to the target data, together with maximizing mutual information.

However, in many practical situations models should perform well on both the target and source domain. For example, we would desire a recognition model deployed in an urban environment which works well for all four seasons (domains) after adapting model to the seasons sequentially. As shown in [47], the source performance of some DA methods will degrade after adaptation even with source data always at hand. And the current SFDA methods focus on the target domain by fine tuning the source model, leading to forgetting on old domains. Thus, existing methods cannot handle the situation described above. A simple way to address this setting is by just storing the source and target model, however, we aim for memory-efficient solutions that scale sub-linear with the number of domains. Therefore, in this paper, we propose a new DA paradigm where the model is expected to perform well on all domains after source-free domain adaptation. We call this setting *Generalized Source-free Domain Adaptation* (G-SFDA). For simplicity, in the paper we will first focus on a single target domain, and then we describe how to extend to Continual Source-free Domain Adaptation.

*Corresponding Author.

In this paper, to perform adaptation to the target domain without source data, we first propose Local Structure Clustering (LSC), that clusters each target feature together with its nearest neighbors. The motivation is that one target feature should have similar prediction with its semantic close neighbors. To keep source performance, we propose to use sparse domain attention (SDA), applied to the output of the feature extractor, activating different feature channels depending on the particular domain. The source domain attention will be used to regularize the gradient during target adaptation to prevent forgetting of source information. With LSC and SDA, the adapted model can achieve excellent performance on both source and target domains. In the experiments, we show that for target performance our method is on par with or better than existing DA and SFDA methods on several benchmarks, specifically achieving state-of-the-art performance on VisDA (85.4%), while simultaneously keeping good source performance. We also extend our method to Continual Source-free Domain Adaptation, where there is more than one target domain, further demonstrating the efficiency of our method.

We summarize our contributions as follows:

- We propose a new domain adaptation paradigm denoted as Generalized Source-free Domain Adaptation (G-SFDA), where the source-pretrained model is adapted to target domains while keeping the performance on the source domain, in the absence of source data.
- We propose local structure clustering (LSC) to achieve source-free domain adaptation, which utilizes local neighbor information in feature space.
- We propose Sparse domain attention (SDA) which activates different feature channels for different domains, and regularizes the gradient of back propagation during target adaptation to keep information of the source domain.
- In experiments, we show that where existing methods suffer from forgetting and obtain bad performance on the source domain, our method is able to maintain source domain performance. Furthermore, when focusing on the target domain our method is on par with or better than existing methods, especially we achieve state-of-the-art target performance on VisDA.

2. Related Works

Here we discuss related domain adaptation settings.

Domain Adaptation. Early domain adaptation methods such as [21, 37, 39] adopt moment matching to align feature distributions. Inspired by adversarial learning, DANN [7] formulates domain adaptation as an adversarial two-player game. CDAN [22] trains a deep networks conditioned on several sources of information. DIRT-T [35] performs domain adversarial training with an added term that penalizes violations of the cluster assumption. Domain adaptation has also been tackled from other perspectives. MCD [31] adopts prediction diversity between multiple learnable classifiers to achieve local or category-level feature alignment between source and target domains. DAMN [3] introduces a framework where each domain undergoes a different sequence of operations. AFN [44] shows that the erratic discrimination of target features stems from much smaller norms than those found in source features. SRDC [38] proposes to directly uncover the intrinsic target discrimination via discriminative clustering to achieve adaptation. The most relevant paper to our LSC is DANCE [29], which is for universal domain adaptation and based on neighborhood clustering. But they are based on instance discrimination [43] between all features, while our method applies consistency regularization on only a few semantically close neighbors.

Source-free Domain Adaptation. Normal domain adaptation methods require access to source data during adaptation. Recently, there are several methods investigating source-free domain adaptation. USFDA [14] and FS [15] explore the source-free universal DA [48] and open-set DA [32], DECISION [2] is for multi-source DA. Related to our work are SHOT [20] and 3C-GAN [18], both for close-set DA. SHOT proposes to fix the source classifier and match the target features to the fixed classifier by maximizing mutual information and pseudo label. 3C-GAN synthesizes labeled target-style training images based on conditional GAN. Recently, BAIT [46] extends diverse classifier based domain adaptation methods to also be applicable for SFDA. Though achieving good target performance, these methods cannot maintain source performance after adaptation. Other than these methods, we aim to maintain source-domain performance after adaptation.

Continual Domain Adaptation. Continual learning (CL) [13, 19, 23, 25] specifically focuses on avoiding catastrophic forgetting when learning new tasks, but it is not tailored for DA since new tasks in CL usually have labeled data. Recently, a few works [4, 26, 36] have emerged that aim to tackle the *Continual Domain Adaptation* (CDA) problem. [4] uses sample replay to avoid forgetting together with domain adversarial training, [26] builds a domain relation graph, and [36] builds a domain-specific memory buffer for each domain to regularize the gradient on both target and memory buffer. Although these methods achieve good performance, they all demand access to source data. And [16] is source-free but they focus on class incremental single target domain adaptation where there is only one-shot labeled target data per class, while our method is related to

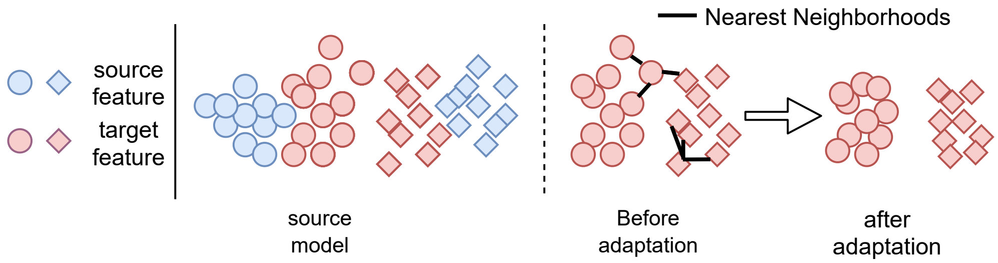


Figure 1: Local Structure Clustering (LSC). Some target features from source model will deviate from dense source feature regions due to domain shift. LSC aims to cluster target features by its semantically close neighbors (linked by black line).

domain incremental learning and can be deployed for continual source-free domain adaptation.

3. Methods

In this section, we first propose an approach for source-free unsupervised domain adaptation. Then we introduce our method to prevent forgetting of the knowledge of the source model. Next, we elaborate how to unify the two modules to address generalized source-free domain adaptation (G-SFDA), and train a domain classifier for domain-agnostic evaluation. Finally, we extend our method to continual source-free domains.

3.1. Problem Setting and Notations

We denote the labeled source domain data with n_s, the samples as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where the y_i^s is the corresponding label of x_i^s, and the unlabeled target domain data with n_t samples as $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$. The number of classes is C. In the source-free setting we consider here $\mathcal{D}_s$ is only available during model pretraining. Our method is based on a neural network, which we split into two parts: a feature extractor f, and a classifier g that only contains one fully connected layer. The output of network is denoted as $p(x) = g(f(x)) \in \mathcal{R}^C$.

3.2. Local Structure Clustering

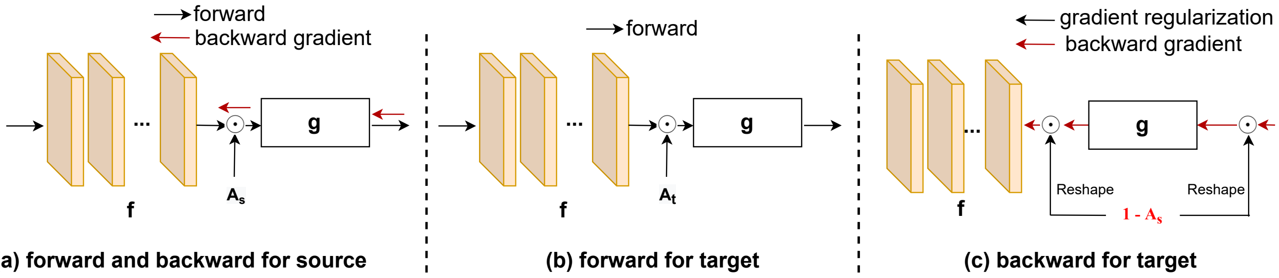
Most domain adaptation methods aim to align the feature distributions of the source and target domain. In source-free unsupervised domain adaptation (SFDA) this is not evident since the algorithm has no longer access to source domain data during adaptation. We identify two main sources of information that the trained source model provides with respect to the target data: a class prediction $p(x)$ and a location in the feature space $f(x)$. The main idea behind our method is that we expect the features of the target domain to be shifted with respect to the source domain, however, we expect that classes still form clusters in the feature space, and as such, we aim to move clusters of data points to their most likely class prediction.

Our algorithm is illustrated in Fig. 1 (left). Some target features (at the start of adaptation) deviate from the corresponding dense source feature region due to domain shift. This could result in wrong prediction of the classifier. However, we assume that the target features of the same class are clustered together. Therefore, the nearest neighbors of target features have a high probability to share category labels. To exploit this fact, we encourage features close in feature space to have similar prediction to their nearest neighbors. As a consequences clusters of points that are close in feature space will move jointly towards a common class. As shown in the right of Fig. 1, this process can correctly classify target features which would otherwise have been wrongly classified.

To find the semantically close neighbors, we build a feature bank $\mathcal{F} = \{(f(x_i))\}_{x_i \in \mathcal{D}_t}$ which stores the target features. This is similar to methods in unsupervised learning [43, 10, 50, 40] or domain adaptation [29]. The method [29] is for universal domain adaptation, and considers similarity based on instance discrimination [43] between all features in their loss function, and [10, 40, 50] perform unsupervised learning using neighborhood information. The work [40] needs pretext training and the nearest neighborhood *images* is retrieved only once by the embedding network from the pretext stage to train another classification network, while [10, 50] are also based on instance discrimination between all target features, and utilize neighbourhood selection to further improve the cluster performance. Different from them, we only use a few neighbors from the feature bank to cluster the target features with a consistency regularization.

Next, we build a score bank $\mathcal{S} = \{(g(f(x_i))\}_{x_i \in \mathcal{D}_t}$ storing corresponding softmaxed prediction scores. The local structure clustering is achieved by encouraging consistent predictions between the k-nearest *features* applying the following loss:

$$\mathcal{L}_{\text{LSC}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} log[p(x_i) \cdot s(\mathcal{N}_k)] + \sum_{c=1}^{C} \text{KL}(\bar{p}_c || q_c)$$
$$\mathcal{N}_{\{1,..,K\}} = \{\mathcal{F}_j |\ top\text{-}K(cos\left(f\left(x_i\right), \mathcal{F}_j\right), \forall \mathcal{F}_j\ \in \mathcal{F})\},$$
$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_c(x_i)\ \text{, and } q_{\{c=1,..,C\}} = \frac{1}{C} \tag{1}$$



(a) forward and backward for source

(b) forward for target

(c) backward for target

Figure 2: (a-c): Forward and Backward pass for two domains. **f**, **g** denote feature extractor, classifier. $\mathcal{A}_s$ and $\mathcal{A}_t$ are the sparse source and target domain attention.

Here, we first find the k-nearest neighbors $\mathcal{N}$ in the feature bank for each current target feature based on the cosine similarity. We minimize the negative log value of the dot product between prediction score of the current target sample x_i and the stored prediction scores $s(\mathcal{N}_k)$ of $\mathcal{N}$, which is the first term in Eq. 1, aiming to encourage consistent predictions between the feature and its a few neighbors. The second term avoids the degenerated solution [34, 8], where the prediction of classes in the target data is highly imbalanced, by encouraging prediction balance. Here p_c is the empirical label distribution; it represents the predicted possibility of class c and q is a uniform distribution. And we simply replace the old items in the bank with the new ones corresponding to current mini-batch. In the experiments, we will prove the effectiveness of the proposed LSC by verifying whether the nearest neighbors are sharing the right predicted label.

3.3. Sparse Domain Attention

Under the G-SFDA setting, we want to not only have high target performance, but maintain source performance without accessing source data. Our work is inspired by continual learning (CL) methods [1, 25, 33] which put constraints on each layer for leaving out capacity for new tasks and prevent forgetting of previous tasks. We propose to only activate parts of the feature channels of $f(x) \in \mathcal{R}^d$ for different domains, by a sparse domain attention (SDA) vector $\mathcal{A}_{i\in\{s,t\}} \in \mathcal{R}^d$, which contain close-to binary values that will mask the output of the feature extractor. Inspired by [33], we adopt an embedding layer to automatically produce the domain adaptation.

$$\mathcal{A}_{\text{i}\in[s,t]} = \sigma(100 \cdot e_i) \tag{2}$$

where e_i is the output of an embedding layer, σ is $sigmoid$ function, and the constant 100 is to ensure a near-binary output, but still differentiable. $\mathcal{A}_s$ and $\mathcal{A}_t$ are both trained on the source domain and are fixed during the adaptation to the target domain. Furthermore, when training on source, we use sparsity regularization and gradient compensation for the embedding layer just like [33]. Thus, we use SDA to build domain specific information flows where some channels are specific for each domain. We can maintain the source information by regularizing the gradient flowing into channels that are activated in the source mask.

For training the source domain, we apply the source attention $\mathcal{A}_s$, as shown in Fig. 2(a), the output is $g(f(x) \odot \mathcal{A}_s)$. In Fig. 2(b), we show that when adapting to the target domain, we use the sparse target attention $\mathcal{A}_t$ for the forward pass. To prevent forgetting, there should be no update to the feature channels which are present in $\mathcal{A}_s$. The reasons are twofold: firstly, the information of those channels is the only source information provided during source-free adaptation to the target domain; keeping this information may boost target adaptation, and secondly more importantly, under the G-SFDA setting we hope to keep the source performance after adapting, therefore target adaptation should not disturb the information flowing to those channels of feature associated with source domain. As shown in Fig. 2(c), during target adaptation we propose to use source attention A_s to regularize the gradients flowing to the classifier and feature extractor during back propagation:

$$W_{f_l} \leftarrow W_{f_l} - (\bar{\mathcal{A}}_s \mathbb{1}_h^T) \odot \frac{\partial \mathcal{L}}{\partial W_{f_l}} \tag{3}$$

$$W_g \leftarrow W_g - \frac{\partial \mathcal{L}}{\partial W_g} \odot (\mathbb{1}_C \bar{\mathcal{A}}_s^T) \tag{4}$$

where $\odot$ denotes element wise multiplication, $\mathbb{1}_k$ is an all-ones vector of dimensionality k, $\bar{\mathcal{A}}_s = 1 - \mathcal{A}_s$, $W_{f_l} \in \mathcal{R}^{d\times h}$ is the weight of the last layer in feature extractor, $W_g \in \mathcal{R}^{C\times d}$ is the weight of the classifier. Here the source attention $\mathcal{A}_s$ is used to regularize the gradient flowing into the source activated channels (for feature extractor) and also the corresponding neurons in the classifier. With Eq. 3 and Eq. 4, the source information is expected to be preserved.

In continual learning literature the masking of weights [24, 25] and activations [1, 27, 33] has been studied. Our method is related to the activation mask methods. However, other then these methods, our masking only prevents forgetting in the last two layers W_{f_l} and W_g. We ensure that the features that are crucial for source domain performance are only minimally changed, and that the target domain specific features are used to address the domain shift. Our approach does not prevent all forgetting of the source

Algorithm 1 Generalized Source-free Domain Adaptation

```
Require: $\mathcal{D}_s$ (only for source model training), $\mathcal{D}_t$
Pre-train model on $\mathcal{D}_s$ with both $\mathcal{A}_s$ and $\mathcal{A}_t$ from SDA
Build feature bank $\mathcal{F}$ and score bank $\mathcal{S}$ for $\mathcal{D}_t$
while Adaptation do
    Sample batch $\mathcal{T}$ from $\mathcal{D}_t$
    Update $\mathcal{F}$ and $\mathcal{S}$ corresponding to current batch $\mathcal{T}$
    Compute $\mathcal{L}_{lsc}$ based on $\mathcal{F}$ and $\mathcal{S}$        ▷ Eq. 1,5
    Update network with SDA regularization        ▷ Eq. 3,4
end while
```

domain, since we do not regularize the gradient of the inner layers in feature extractor.

3.4. Unified Training

In this section, we first illustrate how to unify the training with SDA and LSC. As illustrated in Algorithm 1, first we train the model on $\mathcal{D}_s$ with the cross-entropy loss, with both source and target domain attention $\mathcal{A}_s$, $\mathcal{A}_t$, this is to provide a good initialization for target adaptation where only $\mathcal{A}_t$ is engaged. Then, we adapt the source model to the target domain with target attention $\mathcal{A}_t$ and only access to $\mathcal{D}_t$ with Eq. 1. During backpropagation we regularize the gradients according to Eq. 3 and Eq. 4. Unlike training with only LSC in Sec. 3.2, here we build the feature bank as $\mathcal{F} = \{(f(x_i) \odot \mathcal{A}_t)\}_{x_i \in \mathcal{D}_t}$, where we abandon the irrelevant channels since those channels will not contribute to current prediction and may contain noise. And for the same reason when using k-nearest neighbors, we also apply the target attention to the feature, so the $\mathcal{N}_{\{1,..,K\}}$ in Eq. 1 turns into:

$$\mathcal{N}_{\{1,..,K\}} = \{\mathcal{F}_j | \, \textit{top-K}(cos\,(f\,(x_i) \odot \mathcal{A}_t, \mathcal{F}_j)\,, \forall \mathcal{F}_j \, \in \mathcal{F})\} \tag{5}$$

Domain-ID estimation. In the experimental section, we will consider both G-SFDA with (*domain-aware*) and without (*domain-agnostic*) access to the domain-id at inference time. In the more challenging setting the domain-ID is not available, and needs to be estimated. Therefore, we propose to train a domain classifier which takes in feature $f(x)$ to estimate the domain-ID of the test samples, by only storing a very small set of images of the source domain. We will show in the experiments that we obtain similar results in the challenging domain-agnostic setting as in the easier domain-aware setting.

3.5. Continual Source-free Domain Adaptation

Here we illustrate how to extend our method to continual source-free domain adaptation, where the model is adapted to a sequence of target domains with only access to current target domain data. Assuming that there are N_t target domains. For source pretraining we train with all domain attention $\mathcal{A}_s$ and $\{\mathcal{A}_{t_i}\}_{i=1..N_t}$ from SDA, for a good initialization as mentioned before. And when adapting to the j-th target domain, we compute $\mathcal{A}'$ which considers all domain attention except the current one. We replace the $\mathcal{A}_s$ in Eq. 3 and Eq. 4 with $\mathcal{A}'$ for current gradient regularization:

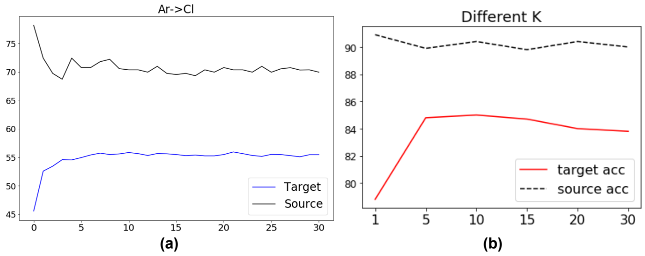


Figure 3: (a) Training curves on task Ar→Cl of Office-Home dataset. (b) Ablation study of different K on VisDA.

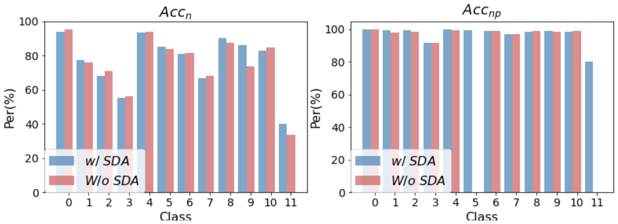


Figure 4: Ablation study of SDA on VisDA, which has 12 classes. Acc_n means the percentage of target features which share the same **predicted** label with its 3 nearest neighbors, and Acc_{np} means the percentage among above features which have the **correct** shared predicted class.

$$\mathcal{A}' = \max(\mathcal{A}', \mathcal{A}_{t_i}), \; \forall i \in \{1, .., N_t\} \setminus j \tag{6}$$

where max is an element-wise operation and $\mathcal{A}'$ is initialized from $\mathcal{A}_s$. Using $\mathcal{A}'$ for gradient regularization means training on one target domain should not influence others.

4. Experiments

Datasets. *Office-Home* [41] contains 4 domains (Real, Clipart, Art, Product) with 65 classes and a total of 15,500 images. *VisDA* [28] is a more challenging dataset with 12 classes. Its source domain contains 152k synthetic images while the target domain has 55k real object images.

Evaluation. We mainly compare with existing methods under two different settings, one is the normal DA and SFDA setting where target performance is the only focus. Another is our proposed G-SFDA setting, where the adapted model is expected to have good performance on both source and target domains after source-free domain adaptation. In this setting, we compute the harmonic mean between source and target accuracy: $H = \frac{2 * Acc_S * Acc_T}{Acc_S + Acc_T}$, and Acc_S and Acc_T are respectively the accuracy on source and target test data. For SFDA, we use all source data for model pretraining. And for G-SFDA we only use part (80% for Office-Home and 90% for VisDA), the remaining source data is used for evaluating source performance. We provide results under both the domain aware and domain agnostic setting (where we estimate

Method (Synthesis → Real)	Source-free	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
ResNet-101 [9]	×	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
ADR [30]	×	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 [22]	×	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 [5]	×	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
SWD [17]	×	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
MDD [49]	×	-	-	-	-	-	-	-	-	-	-	-	-	74.6
IA [11]	×	-	-	-	-	-	-	-	-	-	-	-	-	75.8
DMRL [42]	×	-	-	-	-	-	-	-	-	-	-	-	-	75.5
MCC [12]	×	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
DANCE [29]	×	-	-	-	-	-	-	-	-	-	-	-	-	70.4
DANCE [29]	√	-	-	-	-	-	-	-	-	-	-	-	-	70.2
SHOT [20]	√	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
3C-GAN [18]	√	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
Ours w/ domainID	√	96.1	88.3	85.5	74.1	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	**85.4**

Table 1: Accuracies (%) on VisDA-C for ResNet101-based unsupervised domain adaptation methods. Source-free means setting without access to source data during adaptation. Underlined results are second highest result. Our results are using target attention $\mathcal{A}_t$.

Method	Source-free	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-50 [9]	×	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
MCD [31]	×	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [22]	×	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
MDD [49]	×	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
IA [11]	×	56.0	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
BNM [6]	×	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
BDG [45]	×	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [38]	×	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 [20]	√	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 w/ domainID	√	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3

Table 2: Accuracies (%) on Office-Home for ResNet50-based unsupervised domain adaptation methods. Source-free means source-free setting without access to source data during adaptation. Underline means the second highest result. Our results are using target attention $\mathcal{A}_t$.

		plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Avg.**	
	Source-free	S/T	S/T	S/T	S/T	S/T	S/T	S /T	S /T	S /T	S /T	S /T	S /T	**S /T**	**H**
Source model		99.9/70.6	99.9/15.6	99.3/45.6	99.1/80.9	99.9/63.0	99.9/5.1	99.4/79.2	100/24.9	99.9/64.0	100/39.6	99.3/84.8	98.3/6.3	**99.6** /48.1	64.9
SHOT [20]	√	99.3/94.4	97.3/85.8	34.9/78.4	47.3/55.2	94.4/93.9	93.2/95.0	38.3/81.5	94.4/79.5	99.1/89.8	92.7/90.1	55.4/85.6	62.0/56.8	75.7/82.2	78.8
Ours w/ domain-ID	√	99.7/95.9	98.7/88.1	98.4/85.4	80.0/72.5	94.6/96.1	98.4/93.7	76.2/88.5	97.8/80.6	98.8/92.3	99.9/92.2	75.6/87.6	67.3/44.8	90.4/**85.0**	**87.6**
Ours w/o domain-ID	√	99.7/95.4	98.7/87.7	98.4/85.7	80.0/71.5	94.6/96.1	98.4/94.8	76.2/89.2	97.8/80.4	98.8/92.0	99.9/88.6	75.6/87.4	67.3/44.1	90.4/84.4	87.3

Table 3: Accuracy (%) of each method on VisDA dataset using ResNet-101 as backbone under **G-SFDA** setting. Randomly specifying 0.9/0.1 train/test split for the source dataset. T and S denote accuracy on target and source domain. Domain-ID means having access to domain-ID during evaluation, we provide results under both domain aware and agnostic setting.

		Ar → Cl			Ar → Pr			Ar → Rw			Cl → Ar			Cl → Pr			Cl → Rw		
	Source-free	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H
Source model		78.2	45.0	57.1	78.2	67.2	72.3	78.2	73.9	76.0	79.7	49.0	60.7	79.7	59.7	68.3	79.7	62.2	69.9
SHOT [20]	√	60.9	55.3	58.0	65.2	77.4	70.8	71.6	80.8	75.9	65.9	68.4	67.1	63.5	76.9	69.6	67.4	75.7	71.3
Ours w/ domain-ID	√	70.0	54.9	61.5	74.0	77.1	75.5	74.5	79.7	77.0	78.5	67.0	72.7	80.3	76.1	78.1	80.6	78.4	79.5
Ours w/o domain-ID	√	68.8	54.7	60.9	72.0	75.6	73.8	74.5	78.5	76.4	77.2	66.6	71.5	79.7	74.0	76.7	78.5	78.4	78.4

	Pr → Ar			Pr → Cl			Pr → Rw			Rw → Ar			Rw → Cl			Rw → Pr			**Avg.**		
	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H	**S**	**T**	**H**
Source model	92.3	52.0	66.5	92.3	40.3	56.1	92.3	73.0	81.5	85.4	64.7	73.6	85.4	45.8	59.6	85.4	77.5	81.3	**83.9**	59.2	68.6
SHOT [20]	78.9	65.4	71.5	74.2	54.2	62.6	84.9	80.5	82.6	79.7	71.7	75.5	71.0	59.0	64.4	79.2	84.6	81.8	71.9	**70.8**	70.9
Ours w/ domain-ID	89.8	65.7	75.9	89.3	53.8	67.1	91.6	81.9	86.5	85.9	71.5	78.0	81.3	60.5	69.4	84.4	83.4	83.9	81.8	**70.8**	**75.5**
Ours w/o domain-ID	87.8	65.1	74.8	86.3	53.2	65.8	90.3	81.6	85.7	83.2	72.0	77.2	78.3	60.2	68.1	83.4	82.8	83.1	80.0	70.2	74.4

Table 4: Accuracy (%) of each method on Office-Home dataset using ResNet-50 as backbone under **G-SFDA** setting. Randomly specifying 0.8/0.2 train/test split for the source dataset. T and S denote accuracy on target and source domain. domain-ID means having access to domain-ID during evaluation, w/o domain-ID means using the estimated domain-ID from domain classifier.

Office-Home	S	T
Source model	**83.9**	59.2
Ours (w/o SDA)	72.4	70.2
Ours (w/ SDA)	81.8	**70.8**

VisDA	S	T
Source model	**99.6**	48.1
Ours (w/o SDA)	72.1	74.6
Ours (w/ SDA)	90.4	**85.0**

OH /s	S	T
65 (paper)	80.0	70.2
130	80.6	70.3
195	**80.8**	**70.4**

VisDA /s	S	T
16	89.0	83.6
32	90.2	84.2
64 (paper)	**90.4**	**84.4**

Table 5: (**Left** two) Ablation study on Office-Home and VisDA. The S and T means source and target accuracy. (**Right** two) Ablation on number of stored images per domain to train domain classifier.

	test			
	Ar	Cl	Pr	Rw
Ar	74.5	42.0	61.3	68.2
Cl	71.4	56.6	61.2	67.9
Pr	70.9	55.7	73.0	71.2
Rw	72.6	55.6	72.7	77.2

	test			
	Cl	Ar	Pr	Rw
Cl	82.2	49.7	60.0	61.2
Ar	80.1	65.4	63.7	66.3
Pr	79.7	63.2	72.9	68.2
Rw	78.6	64.9	72.8	72.4

	test			
	Pr	Ar	Cl	Rw
Pr	92.0	49.7	41.0	71.0
Ar	91.0	63.6	42.7	72.6
Cl	89.2	61.8	53.1	70.4
Rw	88.6	63.1	51.5	76.5

	test			
	Rw	Ar	Cl	Pr
Rw	86.0	63.0	45.7	77.6
Ar	85.7	72.4	49.8	77.4
Cl	80.7	68.9	59.1	73.4
Pr	84.2	69.1	57.4	80.5

Table 6: Continual Source-free Domain Adaptation, the model is adapted from source domain (the first domain) to all target domain sequentially. The results on source domain are reported on the test set.

the domain-ID with the domain classifier). Finally, we report results for continual source-free domain adaptation.

Model details. We adopt the backbone of ResNet-50 [9] for Office-Home and 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 64 on all datasets. The learning rate for Office-Home is set to 1e-3 for all layers, except for the last two newly added fc layers, where we apply 1e-2. Learning rates are set 10 times smaller for VisDA. *On the source domain, we train the whole network with all domain attentions from SDA, while for target adaptation, we only train the BN layers and last layer in feature extractor, as well as the classifier*. We train 30 epochs on the target domain for Office-Home while 15 epochs for VisDA. For the number of nearest neighbors (K) in Eq. 1, we use 2 for Office-Home, since VisDA is much larger we set K to 10. All results are the average between three runs with random seeds. For training the domain classifier, we store one image per class for Office-Home (total 130 images for 65 classes, 2 domains), and randomly sample 64 images per domain for VisDA (total 128 images for 12 classes, 2 domains). The domain classifier only contains 2 fc layers.

4.1. Comparing with State-of-the-art

Target-oriented Domain Adaptation. We first evaluate the target performance of our method compared with existing DA and SFDA methods. The results on the VisDA and Office-Home dataset are shown in Tab. 1-2, our results are using target attention $\mathcal{A}_t$. In these tables, the top part (denoted by × in the *source-free* column) shows results for the normal setting with access to source data during adaptation. The bottom one (denoted by $\surd$ in the *source-free* column) shows results for the source-free setting. Our method achieves state-of-the-art performance on VisDA surpassing SHOT by a large margin (2.5%). The reported results clearly demonstrate the efficiency of the proposed method for source-free domain adaptation. Interestingly, like already observed in the SHOT paper, source-free methods outperform methods that have access to source data during adaptation. Our method is on par with existing DA methods on Office-Home, where our method gets the same results as the DA method SRDC [38] and is a little inferior to the SFDA method SHOT (0.5% lower than SHOT).In addition, we show the results of DANCE [29] with and without source data in Tab. 1 which are almost the same. Since both of DANCE and our method are using neighborhood information for adaptation, these results may imply that source data are not necessity when efficiently exploiting the target feature structure.

Generalized Source-free Domain Adaptation. Here we evaluate our method under the G-SFDA setting. Since we leave out part of the source data for evaluation, we need to reproduce current SFDA methods. 3C-GAN [18] did not release code, we therefore only compare with the source-free method SHOT [20] reproduced by ourselves based on the author's code. As shown in Tab. 3-4, first our method (w/ domain-ID) obtains a significantly higher H value improving SHOT by 8.8% on Office-Home and 4.6% on VisDA. The gain is mainly due to superior results on the source dataset, since SHOT suffers from forgetting. Compared with the source model, our method still has a drop of 2.1% and 9.2% lower on Office-Home and VisDA, implying there is still space to explore further techniques to reduce forgetting. We also report the results for domain agnostic evaluation, where we use the domain classifier to estimate domain-ID. As shown in the last row of Tab. 3 and Tab. 4, with the estimated domain-ID, our methods can get similar results compared with the domain aware method, and still report superior H values compared to SHOT.Note there is still source performance degradation, since we only deploy one SDA module before the classifier. The forgetting is caused in the layers inside the feature extractor. One factor is the statistics in the BN layers which will be replaced by the target statistics after adaptation. If we would adapt the BN parameters back to the source domain (by simply doing a forward pass to update BN statistics before evaluation), we found that this leads to a performance gain (0.7% and 1.6% on Office-Home and VisDA respectively) on the source domain.

4.2. Analysis and further experiments

Training curves. As shown in Fig. 3(a), with SDA the source performance during the whole adaptation stage is quite smooth, which proves the efficiency of SDA.

Number of nearest neighbors K. In Fig. 3(b), we show the results with different $K \in \{1, 5, 10, 15, 20, 30\}$ in Eq. 1 on VisDA. Our method is quite robust to the choice of K, only K is 1 results in lower results. We conjecture that only using a single nearest neighbor in Eq.1 maybe noisy if the feature locates in dense regions.

Ablation study of SDA. We show the results of removing the SDA in the left of Tab. 5. As expected removing SDA leads to a large drop in source performance. Unexpected is that removing SDA also deteriorates target performance: a lot on VisDA (10.4↓), and a little for Office-Home (0.6↓). To further investigate it, we check how well LSC works with and without SDA on VisDA in Fig. 4; here Acc_n means the percentage of target features which share the same predicted label with its 3 nearest neighbors, and among those features Acc_{np} means the percentage having the correct shared predicted label. According to the results, LSC can lead to good local structure (most neighbors share the same prediction), however the prediction maybe wrong if removing SDA, this is especially the case for class 5 and 11 which have totally wrong prediction (Acc_{np} is 0). This may imply keeping source information with SDA is helping target adaptation.

Domain classifier. We report results as a function of the number of stored images for training domain classifier (right of Tab. 5). For Office-Home, we ensure at least one image per class. The results show with a small amount of stored images, the learned domain-ID classifier works well.

t-SNE visualization. We visualize the features before and after adaptation, which are already masked by the different domain attentions, the source and target features are expected to cluster independently, just as shown in Fig. 5. The source clusters maintain well after adaptation, and the disordered target features turn into more structured after adaptation. We also visualize features in the shared and specific domain channels. As shown in Fig. 6, features in the shared domain channels cluster together, but features in the specific domain channels are totally separated across domains.

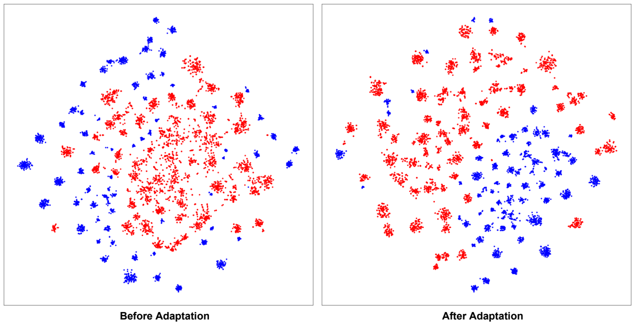


Figure 5: t-SNE visualization of features before and after adaptation on task Ar→Pr of Office-Home. The blue are source features while the red are target.

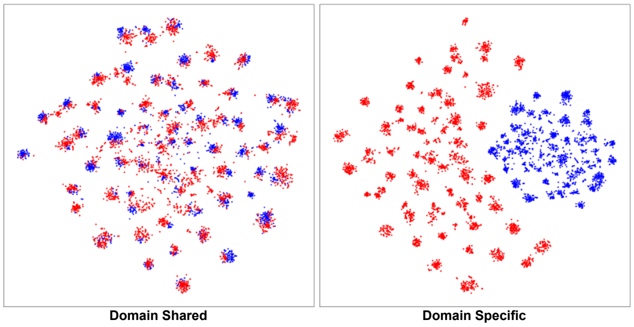


Figure 6: t-SNE of features from domain shared and domain specific channels after adaptation (task Ar→Pr on Office-Home). The blue are source features while red for target.

Continual Source-free Domain Adaptation. We also provide results (domain aware) of continual source-free domain adaptation in Tab. 6. The results show that it can work well for all domains. The interesting thing is that adapting to one target domain will improve the performance on not-seen target domain, for example, when adapting the model from source domain *Cl* to the first target domain *Ar*, the unseen target domain *Rw* also gains. The reason is that the information learned currently is also helpful for future target domain. Note for some target domains, the result is lower compared with directly adapting from source to the domain, the reason is that we decrease the learned channels by using more gradient regularization as in Eq. 6, implying more capacity is needed for adapting to more domains.

5. Conclusion

In this paper, we propose a new domain adaptation paradigm denoted as Generalized Source-free Domain Adaptation, where the learned model needs to have good performance on both the target and source domains, with only access to the unlabeled target domain during adaptation. We propose local structure clustering to keep local target cluster information in feature space, successfully adapting the model to the target domain without source domain data. We propose sparse domain attention, which activates different feature channels for different domains, and is also utilized to regularize the gradient during target training to maintain source domain information. Experiment results testify the efficacy of our method.

Acknowledgement We acknowledge the support from Huawei Kirin Solution, and the project PID2019-104174GB-I00 (MINECO, Spain), and the CERCA Programme of Generalitat de Catalunya.

References

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3940, 2020. 4

[2] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10103–10112, 2021. 2

[3] Roger Bermudez Chacon, Mathieu Salzmann, and Pascal Fua. Domain-adaptive multibranch networks. In *8th International Conference on Learning Representations*, 2020. 2

[4] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. 2018. 2

[5] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1081–1090, 2019. 6

[6] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *CVPR*, 2020. 6

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[8] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745, 2017. 4

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 7

[10] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *International Conference on Machine Learning*, pages 2849–2858. PMLR, 2019. 3

[11] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. *arXiv preprint arXiv:2006.04996*, 2020. 6

[12] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. *ECCV*, 2020. 6

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2

[14] Jogendra Nath Kundu, Naveen Venkat, and R Venkatesh Babu. Universal source-free domain adaptation. *CVPR*, 2020. 1, 2

[15] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, R Venkatesh Babu, et al. Towards inheritable models for open-set domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12376–12385, 2020. 1, 2

[16] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-incremental domain adaptation. *ECCV*, 2020. 2

[17] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019. 6

[18] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 1, 2, 6, 7

[19] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *ICML*, 2020. 1, 2, 6, 7

[21] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *ICML*, 2015. 2

[22] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1647–1657, 2018. 2, 6

[23] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6470–6479, 2017. 2

[24] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018. 4

[25] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 2, 4

[26] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6568–6577, 2019. 2

[27] Marc Masana, Tinne Tuytelaars, and Joost van de Weijer. Ternary feature masks: zero-forgetting for task-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3570–3579, 2021. 4

[28] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain

adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5

[29] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 6, 7

[30] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *ICLR*, 2018. 6

[31] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 2, 6

[32] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–168, 2018. 2

[33] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018. 4

[34] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1275–1282, 2012. 4

[35] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *ICLR*, 2018. 2

[36] Peng Su, Shixiang Tang, Peng Gao, Di Qiu, Ni Zhao, and Xiaogang Wang. Gradient regularized contrastive learning for continual domain adaptation. *AAAI*, 2021. 2

[37] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2

[38] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8725–8735, 2020. 2, 6, 7

[39] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[40] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020. 3

[41] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 5

[42] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. *ECCV*, 2020. 6

[43] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2, 3

[44] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[45] Guanglei Yang, Haifeng Xia, Mingli Ding, and Zhengming Ding. Bi-directional generation for unsupervised domain adaptation. In *AAAI*, pages 6615–6622, 2020. 6

[46] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020. 1, 2

[47] Shaokai Ye, Kailu Wu, Mu Zhou, Yunfei Yang, Sia Huat Tan, Kaidi Xu, Jiebo Song, Chenglong Bao, and Kaisheng Ma. Light-weight calibrator: a separable component for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13736–13745, 2020. 1

[48] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2720–2729, 2019. 2

[49] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019. 6

[50] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019. 3