Separate to Adapt: Open Set Domain Adaptation via Progressive Separation

Hong Liu[1*], Zhangjie Cao[1*], Mingsheng Long[1](✉), Jianmin Wang[1], and Qiang Yang[2]
[1]KLiss, MOE; BNRist; School of Software, Tsinghua University, China
[1]Research Center for Big Data, Tsinghua University, China
[1]Beijing Key Laboratory for Industrial Big Data System and Application
[2]Hong Kong University of Science and Technology, China
h-l17@mails.tsinghua.edu.cn, {mingsheng, jimwang}@tsinghua.edu.cn, qyang@cse.ust.hk

Abstract

Domain adaptation has become a resounding success in leveraging labeled data from a source domain to learn an accurate classifier for an unlabeled target domain. When deployed in the wild, the target domain usually contains unknown classes that are not observed in the source domain. Such setting is termed Open Set Domain Adaptation (OSDA). While several methods have been proposed to address OSDA, none of them takes into account the openness of the target domain, which is measured by the proportion of unknown classes in all target classes. Openness is a critical point in open set domain adaptation and exerts a significant impact on performance. In addition, current work aligns the entire target domain with the source domain without excluding unknown samples, which may give rise to negative transfer due to the mismatch between unknown and known classes. To this end, this paper presents Separate to Adapt (STA), an end-to-end approach to open set domain adaptation. The approach adopts a coarse-to-fine weighting mechanism to progressively separate the samples of unknown and known classes, and simultaneously weigh their importance on feature distribution alignment. Our approach allows openness-robust open set domain adaptation, which can be adaptive to a variety of openness in the target domain. We evaluate STA on several benchmark datasets of various openness levels. Results verify that STA significantly outperforms previous methods.

1. Introduction

Recent development of deep neural networks has improved the performance of diverse computer vision tasks. However, the substantial prerequisite of the performance boost is the access to large amount of annotated training data, which is often prohibitive in many real applications. When labeled data is scarce in the domain of interest, a reasonable alternative is a relevant domain with sufficient supervision.

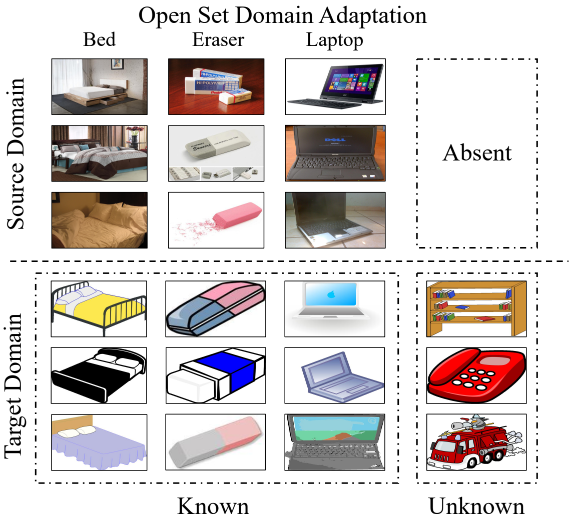


Figure 1. The open set domain adaptation problem, where the target domain contains "unknown" classes absent in the source domain.

We term the domain of interest as the *target domain* and the relevant domain as the *source domain*. However, data from different domains are drawn from different distributions. The domain gap can cause the model to make false predictions in the target domain and degrade the performance [26, 24].

A convincing solution to diminish the distribution shift across domains is domain adaptation. Existing domain adaptation methods seek to bridge domain gap by distribution matching in feature-level [36, 4, 17, 19, 35, 3] or in pixel-level [10, 31, 11, 22, 16]. However, most of previous methods assume that the source and target domains share the same labels, known as *Closed Set Domain Adaptation* [25]. This closed-set setting is still restricted in applications in the wild, since we cannot decide whether source and target domains share the same label space if no target annotations are available. A more realistic setting, *Open Set Domain Adaptation* (OSDA), is therefore recently studied [25, 30, 21]. In this paper, we mainly follow the setting proposed by Saito *et al.* [30], where the target domain has all classes in the source

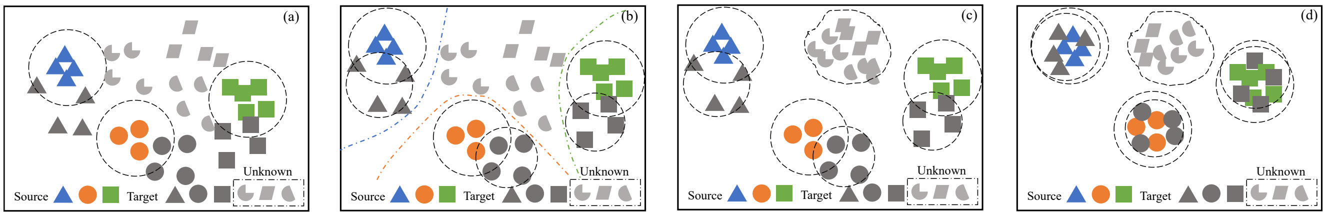


Figure 2. An overview of the proposed Separate to Adapt (STA) approach to open set domain adaptation. Gray shapes are data of target domain and shapes in color are data of source domain. Different kinds of shapes indicate different classes. (a) An example of open set domain adaptation problem, where all source classes are in the target classes and target have unknown classes. (b) The situation after training the multi-binary classifier $G_c|_{c=1}^{|\mathcal{C}_s|}$ for deriving coarse weights to distinguish the unknown classes from known classes in the target domain. Dashed curve in different colors indicates the decision boundary for each binary classifier G_c for the c-th class. It forces the target data of unknown classes to move away from source data. (c) The situation after training the fine-grained binary classifier G_b for deriving more accurate weights. Target data in shared classes and in unknown classes are deviated far away. (d) The situation after the final distribution alignment, where target data in the shared classes are close to their source domain counterparts. *Best viewed in color.*

domain and further contains target-specific classes, as shown in Figure 1. We need to classify data of known classes in the target domain correctly, and reject data of all unknown classes as "unknown" since we have no information about these classes. Open set domain adaptation is more practical, especially for the "in-the-wild" setting where we cannot constrain the boundary of classes in the target domain.

Open set domain adaptation introduces two challenges. (**1**) As presented in classical domain adaptation, it is still essential to mitigate the influence of distribution shift between domains. (**2**) Additionally, aligning the whole distribution of source and target domains as before will be risky since data of unknown classes in the target domain can make performance of domain adaptation model even inferior to a model without adaptation. Such phenomenon is known as negative transfer [24]. Thus in open set domain adaptation, we need to identify the boundary between known and unknown classes as accurately as possible, even without accessible information about the unknown classes. We should further apply adaptation to the known classes in both domains.

Only a few approaches have been proposed to tackle open set domain adaptation. Assign-and-Transform-Iteratively (ATI) [25] studies a distance-based metric to iteratively assign unknown samples. Open Set Back-Propagation (OSBP) further attempts to solve problem with no unknown classes in the source domain. Both approaches require some threshold hyper-parameters to distinguish between known and unknown classes, while setting the hyper-parameters further require prior knowledge on target domain classes. Furthermore, these methods do not take into account the proportion of unknown classes in the target domain, which is termed **openness** [32]. In real world applications, openness can vary drastically and is not accessible before training. Thus, previous methods can be undermined by extreme openness since the dominance of unknown classes leads to difficulty in selecting the hyper-parameters and intensifies negative transfer, validated empirically by experiments in Figure 4. Besides, methods depending on pre-defined hyper-parameters require heavy hyper-parameter selection work. Hence only an openness-robust model equipped with an automatic known/unknown classes separation mechanism can address open set domain adaptation efficiently and effectively.

This paper proposes **Separate to Adapt** (**STA**), an end-to-end approach to tackle open set domain adaptation under various levels of openness. We adopt the domain adversarial learning framework and add one more class to the source classifier for "unknown" class. The main difference between the known and unknown classes lies in that the known classes differ from the source domain only with distribution shift while unknown classes deviate from the source domain much farther with both domain gap and semantic gap. Motivated by this key observation, we develop a progressive separation mechanism consisting of a coarse-to-fine separation pipeline. The first step is training a multi-binary classifier with source data to estimate the similarity between target data and each source class. In the second step, we select the data with extremely high and low similarity as data of known and unknown classes, and train a binary classifier with them to perform fine separation on all target samples. We iterate between the two steps and use instance-level weights to reject samples of unknown classes in adversarial domain adaptation. An overview of STA is given in Figure 2. Experiments regarding different aspects of STA demonstrate that STA outperforms state-of-the-art models on open set domain adaptation datasets. We further demonstrate the STA can work effectively and stably on diverse levels of openness.

2. Related Work

This section briefly reviews works related to ours, including settings of domain adaptation and open set recognition.

Closed Set Domain Adaptation. Closed set domain adaptation methods seek to alleviate performance degradation brought by domain discrepancy. A typical approach is minimizing statistical distances between feature distributions. Deep Adaptation Network (DAN) [17] adds adaptation layers to deep network and minimizes Maximum Mean Discrepancy (MMD) between the kernel embeddings of distributions. Central Moment Discrepancy (CMD) [38] similarly enables domain adaptation by matching only first- and second-order moments. Residual Transfer Network (RTN) [19] improves DAN by adding a shortcut connection and entropy minimization criterion. Joint Adaptation Network (JAN) [20] matches the joint distributions of feature and label of the source and target domains. Maximum Classifier Discrepancy [29] enables domain adaptation by approximating $\mathcal{H}\Delta\mathcal{H}$ distance [1] and minimizing it via feature adaptation. Inspired by Generative Adversarial Nets [6], domain adversarial learning has been introduced to domain adaptation. Domain Adversarial Neural Network (DANN) [4, 5] and Adversarial Discriminative Domain Adaptation (ADDA) [35] employ a domain discriminator to distinguish two domains while the feature extractor is learned to confuse the domain discriminator in a domain adversarial training paradigm. Conditional Domain Adversarial Network (CDAN) [18] improves DANN by matching the joint distributions of labels and features. Unfortunately, closed set domain adaptation methods cannot be applied to open set domain adaptation since they suffer from negative transfer and cannot reject unknown classes.

Open Set Recognition. A vast literature of open set recognition has been conducted to reject outliers while correctly classifying inliers during testing. Scheirer *et al.* [32] proposed a 1-vs-set machine to delineate a decision space from the marginal distance. Open set SVM assigns probabilistic scores to reject unknown samples [13]. They were further improved with compact abating probability models [33]. Bendale *et al.* [2] introduced OpenMax layer to harness deep neural networks for open set recognition. Moreover, Open-Set NN [14] extends upon the Nearest-Neighbor classifier to recognize samples from the unknown class. Note that in open set recognition scenario, there exist outliers that do not belong to the classes in the training dataset. In open set domain adaptation, however, target samples and source samples in the shared classes of both domains further follow different distributions, making the task more challenging.

Open Set Domain Adaptation. This is the umbrella our work falls under. Assign-and-Transform-Iteratively [25] (ATI) exploits distance between the feature of each target sample and the center of each source class to decide whether a target sample belongs to one of source classes or the unknown class. Open Set Back-Propagation (OSBP) [30] trains a feature generator to lead the probability of a target sample to be classified as "unknown" to deviate from the pre-defined threshold. They train their feature extractor and classifier in an adversarial training framework. However, for both of them, problems arise when domain discrepancy is significant or the openness between source and target classes varies in a large range especially to be overly large.

We develop a Separate to Adapt (STA) network to address open set domain adaptation. Our method is robust to a variety of openness levels and does not need manual selection of the threshold parameter between known and unknown classes.

3. Method

In this section, we present an overview of our proposed method, and then describe in detail the training procedure. Figure 3 shows the architecture of STA.

3.1. Open Set Domain Adaptation

In open set domain adaptation (OSDA), we have a *source* domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples and a *target* domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled examples. The source domain is associated with a set of classes $\mathcal{C}_s$, which is shared by the target domain $\mathcal{D}_t$, i.e. $\mathcal{C}_s \subset \mathcal{C}_t$, while the target domain is further associated with a set of additional classes $\mathcal{C}_{t\backslash s}$, all represented by "unknown" since we know nothing about these classes. The source and target domains are sampled from probability distributions p and q respectively. In standard domain adaptation, we have $p \neq q$; and in open set domain adaptation, we further have $p \neq q_{\mathcal{C}_s}$, where $q_{\mathcal{C}_s}$ denotes the distribution of the target domain data belonging to the shared label space $\mathcal{C}_s$. We define the **openness** as $\mathbb{O} = 1 - \frac{|\mathcal{C}_s|}{|\mathcal{C}_t|}$. Note that this definition is in line with the openness introduced in open set recognition [32], since in our scenario, the classes in the source domain are included in the target domain.

Towards open set domain adaptation, this paper presents Separate to Adapt (STA), an end-to-end approach that progressively separates the known classes and unknown classes in the target domain, and simultaneously learns a transferable feature extractor $G_f(\mathbf{x})$ and a classifier $y = G_y(G_f(\mathbf{x}))$ to bridge the cross-domain discrepancy in the shared classes.

3.2. Separate to Adapt

The main challenges of open set domain adaptation include the negative transfer and known/unknown separation. There exists interaction between the two challenges. Negative transfer is the phenomenon that a learner trained with domain adaptation algorithms performs even worse than a classifier trained solely on the source domain. In open set domain adaptation, closed set methods will match the whole target domain with the source domain, thus the unknown classes are also matched with source data. This obvious misalignment causes negative transfer. The solution to negative transfer is only aligning the known classes with the source domain, which exactly gives rise to the second challenge. Therefore, a natural logic of the approach is separating the known and unknown classes in the target domain and performing feature adaptation only on the known-class samples.

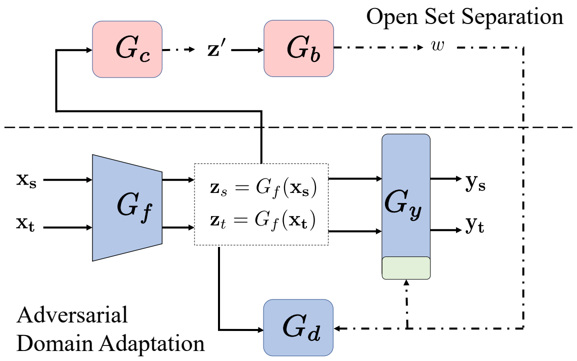


Figure 3. The proposed Separate to Adapt (STA) approach for open set domain adaptation, which is divided into two parts by the dashed line. The top part consists of a multi-binary classifier $G_c|_{c=1}^{|\mathcal{C}_s|}$ and a binary classifier G_b, which will generate the weights w for rejecting target samples in the unknown classes $\mathcal{C}_t \backslash \mathcal{C}_s$. The bottom part consists of feature extractor G_f, classifier G_y and domain discriminator G_d to perform adversarial domain adaptation between source and target data in the shared label space. $\mathbf{z}_s$ and $\mathbf{z}_t$ are the extracted deep features. $\hat{\mathbf{y}}_s$ and $\hat{\mathbf{y}}_t$ are the predicted labels. $\mathbf{z}'$ is the feature selected by G_c. The solid lines show the flow of tensors, and the dashdotted lines indicate the weighting mechanism.

We follow this logic and design our architecture, as shown in Figure 3. It is composed of two parts, where the top part consists of a multi-binary classifier $G_c|_{c=1}^{|\mathcal{C}_s|}$ and a binary classifier G_b to generate weights w for rejecting target samples in the unknown classes $\mathcal{C}_t \backslash \mathcal{C}_s$. The bottom part consists of feature extractor G_f, classifier G_y and domain discriminator G_d to perform adversarial domain adaptation between source and target data in the shared label space $\mathcal{C}_s$.

3.3. Progressive Separation

To separate the data of unknown and known classes in the target domain, we employ a coarse-to-fine filtering process. We utilize a multi-binary classifier, which is composed of $|\mathcal{C}_s|$ binary classifiers denoted by $G_c|_{c=1}^{|\mathcal{C}_s|}$, to measure the similarity between each target sample and each source class. Each binary classifier is trained only with the source data. The loss for all classifiers can be defined as

$$L_s = \sum_{c=1}^{|\mathcal{C}_s|} \frac{1}{n_s} \sum_{i=1}^{n_s} L_{\text{bce}}\left(G_c\left(G_f\left(\mathbf{x}_i^s\right)\right), I\left(y_i^s, c\right)\right), \quad (1)$$

where L_{bce} is the binary cross-entropy loss and $I\left(y_i^s, c\right) = 1$ if $y_i^s = c$ and $I\left(y_i^s, c\right) = 0$ otherwise. Each binary classifier G_c outputs a probability p_c for each target sample to measure how possible the sample belongs to known class c. Thus, the probability p_c can be explained as the similarity between the target sample and known class c. Data of known classes in the target domain tend to have higher probability in one of the shared classes than data of unknown classes. We use the highest probability in $p_1, p_2, ..., p_{|\mathcal{C}_s|}$ as the similarity between each target sample $\mathbf{x}_j^t$ and the source domain:

$$s_j = \max_{c \in \mathcal{C}_s} G_c(G_f(\mathbf{x}_j^t)). \quad (2)$$

With such similarity definition, target data of known classes will have high similarity to their source domain counterparts. Correspondingly, target data of unknown classes will have low similarity to all classes in the source domain.

Thus, we rank the similarity for all the target samples, and choose samples with highest/lowest similarity to train the binary classifier G_b. This filtering is relatively coarse but has high confidence since we only use samples with extreme similarity. It is also robust to different levels of openness since we no longer need to choose hyperparameters manually or using optimization tools.

Another filtering strategy is to cluster the similarities into three clusters for highest, midium and lowest probability respectively. Then we use the mean s_h of the highest probability cluster as the threshold for target data of known class where data with $s_j \geq s_h$ are selected into known classes. And we use the mean s_l of lowest probability cluster as the threshold for data of unknown classes where data with $s_j \leq s_l$ are selected into unknown classes.

With samples selected into known and unknown classes by the multi-binary classifier, we further train a binary classifier G_b to finely separate known and unknown classes. Using $\mathbf{X}'$ to denote the set of filtered samples by the multi-binary classifier, and d_j to indicate whether a target sample $\mathbf{x}_j \in \mathbf{X}'$ is labeled as known ($d_j = 0$) or unknown ($d_j = 1$), the fine-grained binary classifier G_c can be trained as follows,

$$L_b = \frac{1}{|\mathbf{X}'|} \sum_{\mathbf{x}_j \in \mathbf{X}'} L_{\text{bce}}\left(G_b\left(G_f\left(\mathbf{x}_j\right)\right), d_j\right). \quad (3)$$

With the above progressive separation procedure, we can separate data of known and unknown classes in the target domain from coarse ($G_c|_{c=1}^{|\mathcal{C}_s|}$) to fine ($G_b$), thus making the separation more accurate.

3.4. Weighted Adaptation

For the adversarial domain adaptation part, we first define the classification loss of the source domain as follows,

$$L_{\text{cls}}^s = \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y\left(G_y^{1:|\mathcal{C}_s|}\left(G_f\left(\mathbf{x}_i\right)\right), y_i\right), \quad (4)$$

where L_y is cross-entropy loss, G_y is an *extended* classifier for $|\mathcal{C}_s| + 1$ classes, i.e. the $|\mathcal{C}_s|$ known classes in the source domain plus the additional "unknown" class in the target domain. $G_y^{1:|\mathcal{C}_s|}$ denotes the probabilities corresponding to assigning each sample to the $|\mathcal{C}_s|$ known classes.

Then, we need to focus our model on aligning the distributions of source and target data in the shared label space

$\mathcal{C}_s$. Instead of using the output of G_b as a hard discriminator between data of known and unknown classes, we propose to use the softmax output of G_b as a soft instance-level weight, i.e. $w_j = G_b\left(G_f\left(\mathbf{x}_j\right)\right)$, where a larger w_j implies a higher probability to be from the unknown class. Thus, we can exploit w_j to define a weighted loss for adversarial adaptation of the feature distributions in the shared label space $\mathcal{C}_s$ as

$$\begin{aligned} L_d = & \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_{\text{bce}}\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right) \\ & + \frac{1}{\sum\limits_{\mathbf{x}_j \in \mathcal{D}_t} \left(1 - w_j\right)} \sum_{\mathbf{x}_j \in \mathcal{D}_t} \left(1 - w_j\right) L_{\text{bce}}\left(G_d\left(G_f\left(\mathbf{x}_j\right)\right), d_j\right). \end{aligned} \tag{5}$$

In addition, we need to pick out the samples of unknown classes in the target domain to train G_f for the extra "unknown" class. Based on the weight w_j that measures the separation of known and unknown classes, we can define the weighted loss for discriminating the "unknown" class as

$$L_{\text{cls}}^t = \frac{1}{\left|\mathcal{C}_s\right|} \frac{1}{\sum\limits_{\mathbf{x}_j \in \mathcal{D}_t} w_j} \sum_{\mathbf{x}_j \in \mathcal{D}_t} w_j L_y\left(G_y^{\left|\mathcal{C}_s\right|+1}\left(G_f\left(\mathbf{x}_j\right)\right), l_{\text{uk}}\right), \tag{6}$$

where l_{uk} is the unknown class, and through training all target samples with large weights w_j are assigned to the *unknown* class. Similarly, $G_y^{\left|\mathcal{C}_s\right|+1}(G_f)$ is the probability of assigning a target sample to the unknown class by classifier G_y.

We further incorporate the entropy minimization loss L_e on the known classes of target domain to enforce the decision boundary to pass through low-density area in the target domain [7, 19], enhanced by the weights as follows,

$$L_e = \frac{1}{\sum\limits_{\mathbf{x}_j \in \mathcal{D}_t} \left(1 - w_j\right)} \sum_{\mathbf{x}_j \in \mathcal{D}_t} (1 - w_j) H\left(G_y^{1:\left|\mathcal{C}_s\right|}\left(G_f\left(\mathbf{x}_j\right)\right)\right), \tag{7}$$

where H is the entropy loss and $H\left(\mathbf{p}\right) = -\sum_k p_k \log p_k$. It is noteworthy that we only aim to minimize the entropy of target samples estimated to be the known classes, so we use w_j as instance-level weight for the entropy minimization.

3.5. Training Procedure

We divide our training procedure into two steps, a known/unknown separation step and a weighted adversarial adaptation step. We can also alternate between the two steps to progressively adapt samples from the known classes while reject samples from the unknown classes.

Step 1. We first train the feature extractor G_f and the classifier G_y to classify source samples. Meanwhile, the multi-binary classifier $G_c, c = 1, 2, ..., \left|\mathcal{C}_s\right|$ is trained in a one-vs-rest way for each source class. We further select target samples with high/low similarities to the source domain to train the fine-grained binary classifier G_b. Denote by θ_f, θ_y, θ_b, and $\theta_c|_{c=1}^{\left|\mathcal{C}_s\right|}$ the parameters of G_f, G_y, G_b, and $G_c|_{c=1}^{\left|\mathcal{C}_s\right|}$. The optimal parameters $\hat{\theta}_f$, $\hat{\theta}_b$, $\hat{\theta}_y$, and $\hat{\theta}_c|_{c=1}^{\left|\mathcal{C}_s\right|}$ can be found by

$$(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_b, \hat{\theta}_c|_{c=1}^{\left|\mathcal{C}_s\right|}) = \mathop{\arg\min}_{\theta_f, \theta_y, \theta_b, \theta_c|_{c=1}^{\left|\mathcal{C}_s\right|}} L_{\text{cls}}^s + L_s + L_b. \tag{8}$$

Step 2. In this step, we implement adversarial adaptation to align the feature distributions of known classes in the target domain with the source domain, and train G_y for the extra class with data from the unknown class. In this step, we keep training the classifier with source samples to preserve the knowledge from the known classes. Using θ_d to denote the parameters of the domain discriminator G_d, the optimal parameters $\hat{\theta}_f$, $\hat{\theta}_y$, and $\hat{\theta}_d$ can be achieved as follows,

$$(\hat{\theta}_y, \hat{\theta}_d) = \mathop{\arg\min}_{\theta_y, \theta_d} L_{\text{cls}}^s + L_{\text{cls}}^t + L_d + \lambda L_e, \tag{9}$$

$$(\hat{\theta}_f) = \mathop{\arg\min}_{\theta_f} L_{\text{cls}}^s + L_{\text{cls}}^t - L_d + \lambda L_e, \tag{10}$$

where λ is a hyper-parameter to trade off the entropy loss.

With the proposed Separate to Adapt (STA) model, we can efficiently separate the data of the known and unknown classes in the target domain. Step 1 rejects outliers to avoid distraction of unknown classes in Step 2, and Step 2 performs adversarial adaptation to make the rejection pipeline in Step 1 more accurate. Since no threshold hyper-parameters are selected manually in the whole process, we can avoid the painful tuning in real scenarios when the openness $\mathbb{O}$ varies.

4. Experiments

We evaluate the STA model and compare it with state-of-the-art methods in the context of open set domain adaptation. Codes and data will be available at `github.com/thuml`.

4.1. Setup

Office-31 [28] is a standard benchmark for domain adaptation in computer vision with three domains Amazon (**A**), Webcam (**W**) and DSLR (**D**). It contains 4,652 images from 31 categories. We follow previous work [30] using the same set of known classes and unknown classes in the target domain. These tasks represent the performance where the source and target domains have small domain gap.

Office-Home [37] is a challenging domain adaptation dataset, crawled through several search engines and online image directories. It consists of 4 different domains: Artistic (**Ar**), Clipart (**Cl**), Product (**Pr**) and Real-World (**Rw**). Each domain contains images from 65 object classes. We choose (in alphabetic order) the first 25 classes as classes shared by the source and target domains. The 26–65 classes belong to the unknown class. We construct open set domain adaptation tasks between each two domains in both directions, forming 12 tasks where domain discrepancy is substantially larger than Office-31.

Table 1. Classification accuracy (%) of open set domain adaptation tasks on Digits (LeNet) and VisDA-2017 (VGGNet)

	Digits																VisDA-2017								
Method	SVHN → MNIST				USPS → MNIST				MNIST → USPS				Avg				Synthetic → Real								
	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	bicycle	bus	car	motorcycle	train	truck	UNK	OS	OS*
OSVM [13]	54.3	63.1	37.4	10.5	43.1	32.3	63.5	97.5	79.8	77.9	84.2	**89.0**	59.1	57.7	61.7	65.7	31.7	51.6	66.5	70.4	**88.5**	20.8	38.0	52.5	54.9
MMD+OSVM	55.9	64.7	39.1	12.2	62.8	58.9	69.5	82.1	80.0	79.8	81.3	81.0	68.0	68.8	66.3	58.4	39.0	50.1	64.2	79.9	86.6	16.3	44.8	54.4	56.0
DANN+OSVM	62.9	75.3	39.2	0.70	84.4	92.4	72.9	0.90	33.8	40.5	21.4	44.3	60.4	69.4	44.5	15.3	31.8	56.6	**71.7**	77.4	87.0	22.3	41.9	55.5	57.8
ATI-λ	67.6	66.5	69.8	73.0	82.4	81.5	84.0	86.7	86.8	89.6	82.8	73.0	78.9	79.2	78.9	77.6	46.2	57.5	56.9	79.1	81.6	**32.7**	65.0	59.9	59.0
OSBP	63.0	59.1	71.0	82.3	**92.3**	91.2	**94.4**	**97.6**	92.1	**94.9**	88.1	78.0	82.4	81.7	84.5	85.9	51.1	67.1	42.8	84.2	81.8	28.0	**85.1**	62.9	59.2
STA	**76.9**	**75.4**	**80.0**	**84.4**	92.2	**91.3**	93.9	96.5	**93.0**	**94.9**	**90.3**	83.5	**87.3**	**87.2**	**88.1**	**88.1**	**52.4**	**69.6**	59.9	**87.8**	86.5	27.2	84.1	**66.8**	**63.9**

Table 2. Classification Accuracy (%) of open set domain adaptation tasks on Office-31 (ResNet-50)

Method	A → W		A → D		D → W		W → D		D → A		W → A		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
ResNet [9]	82.5±1.2	82.7±0.9	85.2±0.3	85.5±0.9	94.1±0.3	94.3±0.7	96.6±0.2	97.0±0.4	71.6±1.0	71.5±1.1	75.5±1.0	75.2±1.6	84.2	84.4
RTN [19]	85.6±1.2	88.1±1.0	89.5±1.4	90.1±1.6	94.8±0.3	96.2±0.7	97.1±0.2	98.7±0.9	72.3±0.9	72.8±1.5	73.5±0.6	73.9±1.4	85.4	86.8
DANN [4]	85.3±0.7	87.7±1.1	86.5±0.6	87.7±0.6	**97.5**±0.2	**98.3**±0.5	**99.5**±0.1	**100.0**±.0	75.7±1.6	76.2±0.9	74.9±1.2	75.6±0.8	86.6	87.6
OpenMax [2]	87.4±0.5	87.5±0.3	87.1±0.9	88.4±0.9	96.1±0.4	96.2±0.3	98.4±0.3	98.5±0.3	83.4±1.0	82.1±0.6	82.8±0.9	82.8±0.6	89.0	89.3
ATI-λ [25]	87.4±1.5	88.9±1.4	84.3±1.2	86.6±1.1	93.6±1.0	95.3±1.0	96.5±0.9	98.7±0.8	78.0±1.8	79.6±1.5	80.4±1.4	81.4±1.2	86.7	88.4
OSBP [30]	86.5±2.0	87.6±2.1	88.6±1.4	89.2±1.3	97.0±1.0	96.5±0.4	97.9±0.9	98.7±0.6	88.9±2.5	90.6±2.3	85.8±2.5	84.9±1.3	90.8	91.3
STA	**89.5**±0.6	**92.1**±0.5	**93.7**±1.5	**96.1**± 0.4	**97.5**±0.2	96.5±0.5	**99.5**±0.2	99.6±0.1	**89.1**±0.5	**93.5**±0.8	**87.9**±0.9	**87.4**±0.6	**92.9**	**94.1**

VisDA-2017 has two domains where the **Synthetic** one consists of 152,397 synthetic 2D renderings of 3D objects and the **Real** one consists of 55,388 real images. They have 12 classes in common. We follow [30] to construct open set domain adaptation task. This setting validates the efficacy of STA on large-scale synthetic-to-real learning tasks.

Digits have three standard digit classification datasets: MNIST [15], USPS [12] and SVHN [23]. Each dataset contains digits, ranging from 0 to 9. As previous works [30], we construct three open set domain adaptation tasks: **SVHN → MNIST**, **MNIST → USPS** and **USPS → MNIST**.

Caltech-ImageNet is constructed from ImageNet-1K [27] and Caltech-256 datasets. We fix the 84 common classes as known classes and vary unknown classes from 0–916 to testify the robustness against various openness levels.

We compare STA with several open set recognition, domain adaptation and open set domain adaptation methods as previous work [30]: Open Set SVM (**OSVM**) [13], **DANN** [4], **RTN** [19], **OpenMAX** [2], **ATI-λ** [25], **MMD+OSVM**, **DANN+OSVM**, **ATI-λ+OSVM**, and **OSBP** [30]. OSVM is an SVM based approach using thresholding for each class to recognize samples and reject outliers. MMD+OSVM and DANN+OSVM are two variants of OSVM incorporating Maximum Mean Discrepancy [8] and domain adversarial network [4] in OSVM. OpenMax is a deep open set recognition method with a module designed for outlier rejection. ATI-λ maps the feature space of the source domain to the target domain by assigning images in the target domain to known categories. In our setting, there are no source-specific classes. Therefore we manually choose the hyper-parameter λ of ATI-λ by cross-validation. OSBP is the most recent open set domain adaptation method achieving state-of-the-art performance with an adversarial classifier to handle samples of unknown classes. For closed-set methods, we use a confidence threshold to decide whether a sample is from unknown classes. In our experiments, we run each method three times and report average accuracy. Standard deviation of Table 1 is omitted due to the limit of space.

Following previous works [25, 30], we employ four evaluation metrics: **OS:** normalized accuracy for all the classes including the unknown as one class; **OS*:** normalized accuracy only on known classes; **ALL:** the accuracy of all instances (without averaging accuracy over the classes); and **UNK:** the accuracy of unknown samples. We adopt the same experiment setting on Digits and VisDA-2017 datasets as OSBP [30] for fair comparison. We also study STA and all comparing methods on Office-31 dataset with ResNet-50 as the backbone. To further investigate the efficacy of STA with larger domain gap and openness, we study the OS accuracy of all the methods on Office-Home and Caltech-ImageNet datasets with ResNet-50 as the backbone.

For the non-digit datasets, we train the proposed STA models with backbone network VGGNet [34] and ResNet-50 [9] pre-trained on ImageNet [27]. For digit datasets, we use LeNet [15] and train the model from scratch. The domain adversarial network is the same as DANN [4]. All layers trained from scratch have learning rate 10 times that of the pre-trained layers. We use momentum SGD with learning rate searched in a grid range of 10^{-3} to 1 by cross-validation, the momentum is set as 0.9 and the weight decay as 0.0005.

4.2. Results

As shown in Table 1, STA outperforms previous open set methods on **Digits** dataset with different evaluation metrics. Note that, on task **SVHN → MNIST** with larger domain gap, STA improves OSBP by a large margin, which further proves the effectiveness of STA under large domain gap.

We further compare STA with previous methods on the

Table 3. Classification accuracy OS (%) of open set domain adaptation tasks on Office-Home (ResNet-50)

Method	Ar → Cl	Pr → Cl	Rw → Cl	Ar → Pr	Cl → Pr	Rw → Pr	Cl → Ar	Pr → Ar	Rw → Ar	Ar → Rw	Cl → Rw	Pr → Rw	Avg
ResNet [9]	53.4±0.4	52.7±0.6	51.9±0.5	69.3±0.7	61.8±0.5	74.1±0.4	61.4±0.6	64.0±0.3	70.0±0.3	78.7±0.6	71.0±0.6	74.9±0.9	65.3
ATI-λ [25]	55.2±1.2	52.6±1.6	53.5±1.4	69.1±1.1	63.5±1.5	74.1±1.5	61.7±1.2	64.5±0.9	70.7±0.5	79.2±0.7	72.9±0.7	75.8±1.6	66.1
DANN [5]	54.6±0.7	49.7±1.6	51.9±1.4	69.5±1.1	63.5±1.0	72.9±0.8	61.9±1.2	63.3±1.0	71.3±1.0	80.2±0.8	71.7±0.4	74.2±0.4	65.4
OSBP [30]	56.7±1.9	51.5±2.1	49.2±2.4	67.5±1.5	65.5±1.5	74.0±1.5	62.5±2.0	64.8±1.1	69.3±1.1	80.6±0.9	74.7±2.2	71.5±1.9	65.7
OpenMax [2]	56.5±0.4	52.9±0.7	53.7±0.4	69.1±0.3	64.8±0.4	74.5±0.6	**64.1**±0.9	64.0±0.8	71.2±0.8	80.3±0.8	73.0±0.5	76.9±0.3	66.7
STA	**58.1**±0.6	**53.1**±0.9	**54.4**±1.0	**71.6**±1.2	**69.3**±1.0	**81.9**±0.5	63.4±0.5	**65.2**±0.8	**74.9**±1.0	**85.0**±0.2	**75.8**±0.4	**80.8**±0.3	**69.5**

Table 4. Classification accuracy (%) of STA and its three variants on Office-31 (ResNet-50)

Method	A → W		A → D		D → W		W → D		D → A		W → A		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
STA w/o w	87.5±1.4	91.4±1.1	83.0±1.2	89.6±1.2	96.2±0.9	97.3±0.4	98.1±0.7	**100.0**±.0	80.3±1.5	79.3±1.5	71.2±1.2	74.3±1.2	86.1	88.7
STA w/o c	**90.4**±1.7	90.6±1.7	91.5±1.4	91.3±1.4	95.9±1.0	96.7±1.1	98.8±0.6	98.7±0.5	87.4±1.5	87.8±1.5	84.6±1.7	85.2±1.7	91.5	91.8
STA w/o b	85.0±1.5	89.0±1.5	90.6±1.2	91.5±1.3	94.8±1.9	**97.6**±0.8	96.2±0.6	98.2±0.5	77.7±2.2	82.5±2.4	78.9±2.6	83.6±3.5	87.2	90.4
STA w/o j	89.0±1.3	**92.8**±1.2	**94.8**±1.5	95.9±1.0	96.4±0.6	96.2±0.3	98.8±0.7	99.4±0.2	**89.7**±1.4	**93.6**±1.4	85.1±1.1	86.7±1.1	92.5	93.9
STA	89.5±0.6	92.1±0.5	93.7±1.5	**96.1**± 0.4	**97.5**±0.2	96.5±0.5	**99.5**±0.2	99.6±0.1	89.1±0.5	93.5±0.8	**87.9**±0.9	**87.4**±0.6	**92.9**	**94.1**

challenging **VisDA-2017** dataset. STA achieves better performance on most classes, verifying that STA works well with large-scale dataset and very large domain gap between synthetic data and real images.

Results on the six tasks of **Office-31** dataset are shown in Table 2. STA outperforms all comparison methods on most tasks. In particular, we observe that closed set domain adaptation methods perform even worse than ResNet on some tasks. Even with confidence thresholding, these methods cannot work well for open set domain adaptation scenarios. The performance sacrifice comes from the negative transfer caused by wrongly matching unknown classes in the target domain to known classes in the source domain.

Office-Home is challenging in large domain gap and disjoint label space between the source and target domains. From Table 3, we observe that STA exceeds the performance of existing methods by large margins on most tasks. In addition, we observe that previous open set domain adaptation methods perform even worse than the ResNet backbone on some tasks, since they all suffer from the negative effect of unknown classes on domain adaptation. Huge gaps across domains and label spaces aggravate the negative transfer issue brought by the unknown classes and further degrade the performance drastically. STA separates samples of unknown classes before distribution matching, and is thus robust to large domain gap and label-space discrepancy.

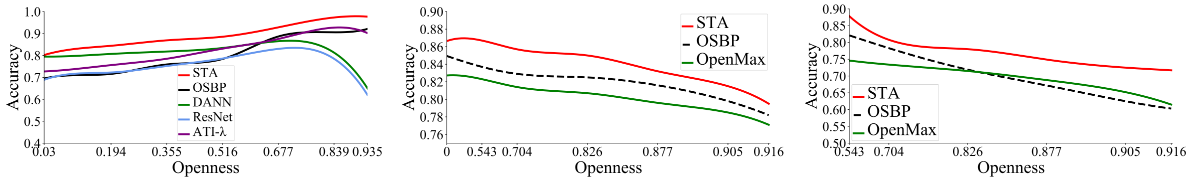
4.3. Analysis

Ablation Study. We compare STA with the variants of STA in Table 4 on **Office-31** dataset. (**1**) STA outperforms **STA w/o w**, the variant without weighting target samples in domain adversarial learning, indicating that aligning samples of unknown classes with source samples leads to negative transfer and performance degradation. Therefore, the weighting we adopt to separate samples of known and unknown classes is necessary. (**2**) Compared with **STA w/o c** which replaces the multi-binary classifier with a softmax classifier, STA achieves significant performance gains, demonstrating multi-binary classifier can generate better similarities to measure the relationship of a target sample to each source class independently. (**3**) STA improves over STA without binary classifier G_b (**STA w/o b**), indicating that the binary classifier can refine the separation between samples of unknown and known classes based on the results of the multi-binary classifier. (**4**) **STA w/o j** is the variants without alternation between the two steps. STA improves over **STA w/o j**, validating the effectiveness of joint separation and adaptation.

Openness. To verify that STA is robust to different levels of openness, we conduct experiments on **Office-31** dataset with openness $\mathbb{O}$ ranging from 0 to nearly 1 (1 is trivial with no target known classes). As shown in Figure 4, previous open set domain adaptation methods perform well only when openness $\mathbb{O}$ is around 0.5. Performance degrades drastically with openness approaching 0 or 1 because those methods are prone to confounding known with unknown classes. ATI-λ and OSBP are able to alleviate the impact of openness to some extent by changing their threshold hyper-parameters λ and t, but this relies on prior knowledge on the openness before training, which is usually unrealistic in real-world applications. With ranking mechanism to configure the multi-binary classifier, STA is robust to openness change without requiring any prior on target domain classes, and therefore performs steadily when the openness varies. In addition, we notice that when openness is close to 0, the performance of STA is still better than DANN. This indicates that the separation mechanism can even filter out noisy target samples in the known classes.

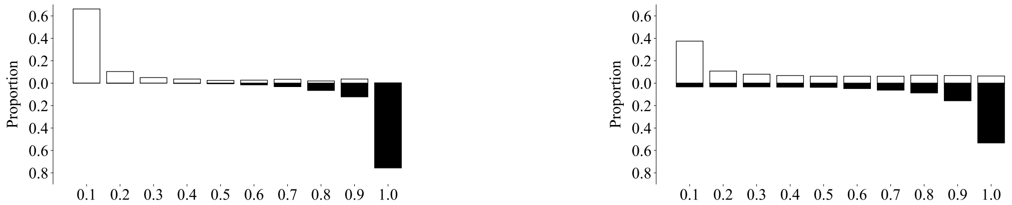
We also conduct experiments with huge variation of openness on **Caltech-ImageNet**, which is close to real-world settings with many unknown classes. Results are shown in Figure 4. We observe that STA exceeds previous methods by large margins on known-class samples and reject outliers accurately under all the openness levels.

Weight Quality. In Figures 5(a) and 5(b), we investigate



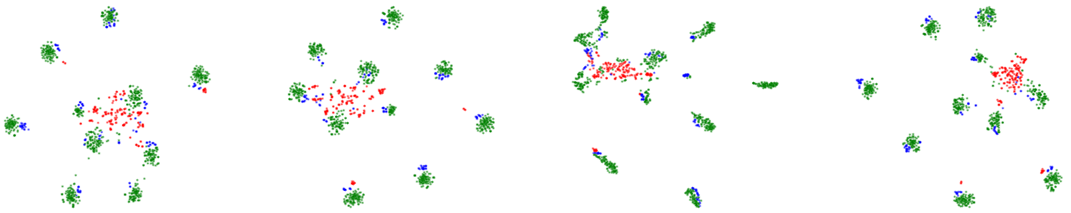
(a) Office-31 OS (b) Caltech-ImageNet Known (c) Caltech-ImageNet Unknown

Figure 4. Accuracy (OS) w.r.t. different openness levels in the target domain.



(a) $\mathbf{A} \rightarrow \mathbf{W}$ (Office-31) (b) VisDA-2017

Figure 5. The w by G_b on (a) $\mathbf{A} \rightarrow \mathbf{W}$ and (b) VisDA-2017. White bins denote target samples of known classes and black unknown classes.



(a) ResNet (b) DANN (c) OSBP (d) STA

Figure 6. Visualization of the features extracted by ResNet, DANN, OSBP, and STA on task $\mathbf{A} \rightarrow \mathbf{D}$ using t-SNE embeddings, respectively. Green points are source features, and blue points are target features of known classes, and red points are target features of unknown classes.

the proportion of samples w.r.t. weight w (the output of the known/unknown binary classifier G_b) on task $\mathbf{A} \rightarrow \mathbf{W}$ and **VisDA-2017** respectively. White bins stand for samples of known classes and black bins for unknown classes, both from the target domain. When the source domain and the target domain are similar (Amazon and Webcam), the outputs of unknown classes are almost 1 while those of known classes are almost 0, indicating that STA perfectly separates target samples into known and unknown classes. Furthermore, when domain discrepancy is huge (Synthetic to Real), we observe from Figure 5(b) that STA can still divide target samples of known and unknown classes effectively.

Feature Visualization. We visualize the last-layer features in ResNet, DANN, OSBP and STA on task **Amazon** $\rightarrow$ **DSLR** in Figure 6(a) to Figure 6(d). We can observe that features of unknown classes and several known classes are close or even mixed together, indicating that ResNet and DANN cannot discriminate known and unknown classes during training. In addition, DANN aligns features of source samples with all target samples and suffers from negative transfer. In Figure 6(c), features of known and unknown classes are drawn apart to some extent, but features of target known classes are not well classified because the adversarial layers in OSBP performs unsteadily when source and target domains are extremely imbalanced. As shown in Figure 6(d), STA is capable of aligning target features of known classes to source features accurately while features of unknown classes are separated far apart even under big openness ($\mathbb{O} = 0.677$).

5. Conclusions

In this paper we address the key challenge in open set domain adaptation, openness, with a novel Separate to Adapt (STA) model. The model clearly separates samples of unknown and known classes in a progressive mechanism, and matches features of known-class samples across source and target domains. As validated on various benchmark datasets, the model enables openness-robust open set domain adaptation under diverse domain discrepancy and disjoint classes.

Acknowledgements

This work is supported by National Key R&D Program of China (No. 2016YFB1000701) and National Natural Science Foundation of China (61772299, 71690231, and 61672313).

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[2] A. Bendale and T. E. Boult. Towards open set deep networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572, June 2016.

[3] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[4] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2017.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[7] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 529–536, 2004.

[8] Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 513–520. MIT Press, 2007.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 1994–2003, 2018.

[11] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[12] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[13] Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409, 2014.

[14] Pedro R. Mendes Júnior, Roberto M. De Souza, Rafael De O. Werneck, Bernardo V. Stein, Daniel V. Pazinato, Waldir R. De Almeida, Otávio A. B. Penatti, Ricardo Da S. Torres, and Anderson Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):1–28, 2016.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015.

[18] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional domain adversarial network. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

[19] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 136–144, 2016.

[20] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning (ICML)*, pages 2208–2217, 2017.

[21] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 165–177. Curran Associates, Inc., 2017.

[22] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[24] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[25] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[26] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[28] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *The European Conference on Computer Vision (ECCV)*, 2010.

[29] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[30] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[31] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, July 2013.

[33] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, Nov 2014.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR, 2015 (arXiv:1409.1556v6)*, 2015.

[35] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. 2014.

[37] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *International Conference on Learning Representations (ICLR)*, May 2017.