Learning to Generate the Unknowns as a Remedy to the Open-Set Domain Shift

Mahsa Baktashmotlagh*
University of Queensland
Australia
m.baktashmotlagh@uq.edu.au

Tianle Chen*
University of Queensland
Australia
tianle.chen@uq.edu.au

Mathieu Salzmann
EPFL
Switzerland
mathieu.salzmann@epfl.ch

Abstract

In many situations, the data one has access to at test time follows a different distribution from the training data. Over the years, this problem has been tackled by closed-set domain adaptation techniques. Recently, open-set domain adaptation has emerged to address the more realistic scenario where additional unknown classes are present in the target data. In this setting, existing techniques focus on the challenging task of isolating the unknown target samples, so as to avoid the negative transfer resulting from aligning the source feature distributions with the broader target one that encompasses the additional unknown classes. Here, we propose a simpler and more effective solution consisting of complementing the source data distribution and making it comparable to the target one by enabling the model to generate source samples corresponding to the unknown target classes. We formulate this as a general module that can be incorporated into any existing closed-set approach and show that this strategy allows us to outperform the state of the art on open-set domain adaptation benchmark datasets.

1. Introduction

Domain shift, referring to the training (i.e., source) and test (i.e., target) data being drawn from different distributions, challenges the standard machine learning assumption [4], thus typically causing dramatic training-testing performance drops. Domain adaptation (DA) aims to alleviate this problem by reducing the gap between the source and target distributions [39, 30, 3, 12, 25, 23]. A particularly popular approach to doing so was inspired by Generative Adversarial Networks [13], and involves the use of an adversarial domain classifier. This classifier attempts to discriminate the source and target features, while the feature extractor aims to fool the discriminator. Many state-of-the-art DA techniques have built on this idea [11, 38, 24, 28] and have proven to be effective at mitigating the domain shift.

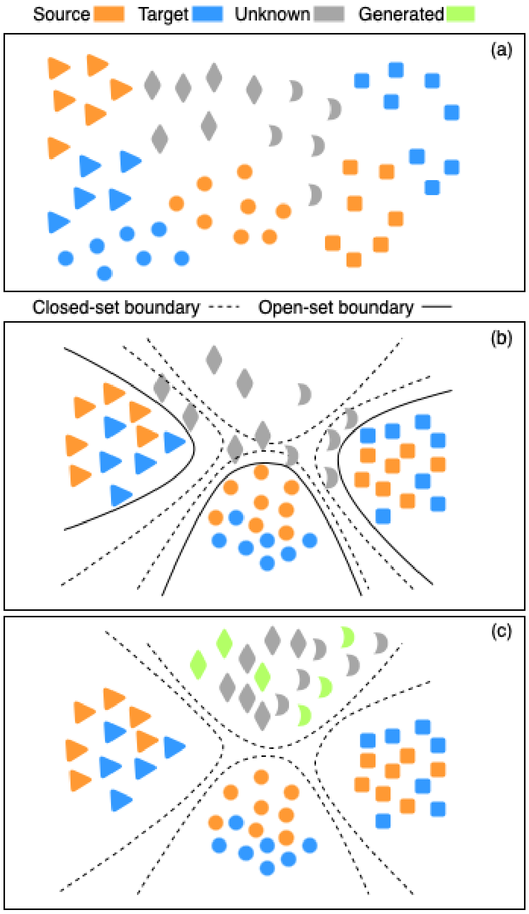


Figure 1: Illustration of our approach. Given source samples from known classes and target samples from both known and unknown classes (a), existing open-set DA methods (b) aim to adjust the decision boundaries to identify the unknowns. By contrast, our approach (c) generates unknown source samples so as to turn the open-set DA into a closed-set one.

Nevertheless, most existing DA techniques tackle the closed-set DA scenario, where the source and target data contain the same classes. As such, they cannot handle

[*] Equal Contribution.

the presence of additional, unknown classes in the target domain, which may further accentuate negative transfer by increasing the source-target distribution mismatch. This more realistic, yet more challenging, scenario is addressed by open-set DA. In this context, the existing methods aim to separate the unknown classes from the known ones, so that distribution alignment can focus on the latter [7, 2, 35, 9, 22]. While isolating the unknown target classes seems intuitive, the resulting methods have to rely on either costly alternative optimization strategies [7], carefully-tuned hyperparameters [2, 35] whose effectiveness highly depends on the openness of the dataset, i.e., the ratio of unknowns to all target samples, or a classifier trained on the source data [22], which may lead to negative transfer when the source and target distributions differ significantly.

In this paper, we introduce a simpler yet more effective approach to open-set DA. Specifically, we propose to complement the source data by generating source samples depicting the unknown target classes so as to reduce the negative transfer entailed by these classes. This is achieved by incorporating a generator that produces unknown source samples into a DA model. To encourage the generated samples to truly encode unknown target classes, we align the distributions of the target and *augmented* source data, while training the final multi-class classifier to account for an *unknown* class, so that the generated samples differ from those containing known classes.

As illustrated in Fig. 1, by generating unknown source samples, we turn open-set DA into a closed-set problem. As such, our solution can be implemented in most existing closed-set DA techniques. The resulting framework outperforms the state-of-the-art open-set DA methods on the challenging Office-Home [40], VisDA-17 [32] and Syn2Real-O [33] benchmarks. In contrast to the existing open-set DA methods, our approach is robust to openness without any hyperparameter tuning. We will make our code publicly available upon acceptance of the paper.

2. Related Work

Closed-set Domain Adaptation: By aiming to mitigate the domain shift between the source and target data, domain adaptation is broadly applicable to many areas, such as computer vision, speech and natural language processing, and robotics. Recent DA approaches can be roughly divided into two categories: statistically-inspired methods [30, 3, 12, 39, 23, 42], which reduce the domain gap by directly minimizing a distribution discrepancy measure between the source and target domain in feature space, and domain-adversarial methods [11, 38, 26, 24, 28], which are motivated by GANs [13] and indirectly align the feature distributions by exploiting a domain discriminator.

Whether statistically inspired or domain adversarial, DA has recently been shown to benefit from the use of pseudo-labels in the target domain [34, 8, 44, 43]. In essence, this strategy consists of labeling a portion of the target samples with the source classifier and using such pseudo-labels as supervision. This process can be performed recursively.

In any event, while the aforementioned unsupervised DA approaches represent great progress in the field, they all tackle the closed-set scenario, where the source and target data contain the same classes. As such, they are vulnerable to the presence of previously-unseen, unknown classes in the target data, which lead to negative transfer.

Open-set Domain Adaptation: While open-set recognition has been relatively well studied in the single-domain scenario [29, 19, 36, 17, 5], the open-set DA literature remains sparse. Assign-and-Transform-Iteratively (ATI) [7] constitutes the first attempt at tackling this challenging, yet more realistic scenario. To this end, it follows an approach similar to pseudo-labeling, assigning the target samples to one of the known or unknown classes based on the distance of the target features to the source class centroids. By contrast, Factorized Representations for Open-set Domain Adaptation (FRODA) [2] separates the known and unknown samples by factorizing them into shared and private representations. Open Set Domain Adaptation by Back Propagation (OSBP) [35] employs a domain adversarial approach, relying on a pre-defined threshold to identify the unknown samples from the known ones. [9] extend OSBP by exploiting a contrastive-center loss to preserve the discriminative information in the known classes while pushing the unknown samples away from the decision boundary. Separate To Adapt (STA) [22] alleviates the need for a pre-defined threshold by exploiting a classifier that estimates the probability of a target sample to belong to one of the source classes or to the unknown ones. Rotation-based Open Set (ROS) [6] exploits the self-supervised task of rotation recognition to align the source and target domains and separate known samples from the unknown ones in the target domain. Self-Ensembling with Category-agnostic Clusters (SE-CC) [31] generalizes the Self-Ensembling technique [10] by using category-agnostic clusters in the target domain, which provide domain-specific visual and structural cues. Inheritable Models for Open-Set Domain Adaptation (InheriTune) [20] defines an objective measure of inheritability to select the most suitable source-trained model, which facilitates adaptation in the absence of the source domain. Progressive Graph Learning (PGL) [27] introduces an end-to-end framework with episodic training to minimize the conditional shift between the source and target distributions.

While promising, the existing open-set DA methods rely on either complex architectures or optimization strategies, or hyper-parameters that make them sensitive to the openness of the dataset, i.e., the ratio of unknowns to all target samples. This is due to the fact that they aim to solve the challenging problem of explicitly isolating the unknown target samples. Here, by contrast, we propose to embrace the presence of unknown classes, and generate unknown source samples to turn the open-set problem into a closed-set one, thus building on the advances of the more mature closed-set DA field.

Note that our approach is different in nature from the ones that use generative models for data augmentation [1] and few-shot learning [41, 15]. Specifically, the former [1] aim to generate samples of observed, known classes, and do not tackle the domain shift problem, and the latter [41, 15] work under the assumption of having access to a few *labeled* images of the new classes, which lets them explicitly focus on the given samples from this class to generate new images, while transferring the modes of variations, e.g., different poses and surroundings, from the base classes. By contrast, we work with two different domains, do not know which target images depict new classes, i.e., the unknown classes are mixed with the known ones, and have access to no labeled target images.

3. Our Approach

Let us now introduce our approach to open-set domain adaptation. To this end, let $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ denote the set of n_s labeled source samples, where $y_i^s \in \mathcal{Y}_s = \{1, \ldots, C-1\}$ is a label coming from one of the $C-1$ known classes. Furthermore, let $\mathcal{D}_t = \left\{\mathbf{x}_j^t\right\}_{j=1}^{n_t}$ denote the set of n_t unlabeled target samples, where $\mathbf{x}_j^t \in \mathcal{X}_t$. Our goal is to learn a classifier $F : \mathcal{X}_t \rightarrow \mathcal{Y}_t$ that, given a target sample $\mathbf{x}^t$, produces a label $\hat{y}^t \in \mathcal{Y}_t = \{1, \ldots, C-1, C\}$, where C jointly accounts for additional, unknown classes, not observed in the source data.

To this end, as depicted by Figure 2, we propose to incorporate a generator network G that, given a noise vector z as input, produces a source sample $\mathbf{x}^g$ from an unknown target class. Our generator consists of six deconvolution layers, with 512, 256, 128, 64, 32, and 3 channels, respectively. These layers use kernels of size 4 and are connected by batch normalization and ReLU nonlinearities. They map an embedding vector of size 100 to an image of size 3 × 224 × 224. For the generated samples to be effective and contain useful information for our underlying open-set DA problem, they must satisfy two properties. First, they must be correctly classified to class C so as to avoid confusion with the known classes. Second, once processed by a feature-extractor backbone network, the data obtained by combining the generated samples with the original source samples must follow the same distribution as the target data. Below, we discuss our approach to enforcing these two properties.

For the first one, let θ_G denote the parameters of the generator $G(z)$, θ_F those of the feature-extractor backbone network $F(\mathbf{x})$, and θ_H those of a multi-class classifier $H(\mathbf{f})$ acting on the features $\mathbf{f}$ computed by the backbone. Our goal then is to learn these parameters so as to solve the problem

$$\min_{\theta_G, \theta_F, \theta_H} L_h(\theta_G, \theta_F, \theta_H) \ , \tag{1}$$

where L_h is

$$\frac{1}{n_s + n_g}\left(\sum_{i=1}^{n_s} L\left(H\left(F\left(\mathbf{x}_i^s\right)\right), y_i^s\right) + \sum_{i=1}^{n_g} L\left(H\left(F\left(\mathbf{x}_i^g(\theta_G)\right)\right), C\right)\right) , \tag{2}$$

with n_g the number of generated samples, and $L(\cdot)$ the cross-entropy loss function.

Solving (1) is of course not sufficient, because it does not exploit the target data at all, and thus cannot encode the second property, i.e., the fact that the distribution of the augmented source data should match that of the target data. To model this, we note that, by augmenting the source data with unknown samples, we have in essence turned open-set DA into a closed-set problem. Therefore, we can exploit the same distribution-alignment strategies as in closed-set DA. Below, we discuss the two most popular such strategies, which we used in our experiments. Note, however, that our formalism extends to most closed-set DA techniques.

Distribution alignment with an adversarial domain classifier. In the context of deep closed-set DA, one of the most popular trends to minimize the discrepancy between the source and target distributions, introduced by [11], consists of jointly training a binary domain classifier $D(\mathbf{f})$. The goal then becomes learning a feature representation that fools this classifier, i.e., makes the target features indistinguishable from the source ones. In our context, and combining this idea with the previous loss function, this can be expressed as the minimax problem

$$\begin{aligned} &\min_{\theta_G, \theta_F, \theta_H} L_h(\theta_G, \theta_F, \theta_H) - \lambda_d L_d(\theta_G, \theta_F, \theta_D) \\ &\min_{\theta_D} \ L_d(\theta_G, \theta_F, \theta_D) \ , \end{aligned} \tag{3}$$

where θ_d denotes the discriminator parameters, λ_d trades off the influence of the two loss terms in the first optimization problem, and $L_d(\cdot)$ is a discriminator loss. As shown by [11], both optimization problems can be solved jointly using a gradient reversal layer. Note that, w.l.o.g., we assume that source samples to be ordered, the original ones first followed by the generated ones. Furthermore, $\mathbf{f}_i^{s,g}$ denotes the feature vector of either an original source sample or a generated one.

In [11], the discriminator loss is the binary cross-entropy defined as

$$L_b = -\frac{1}{n_s + n_g}\sum_{i=1}^{n_s+n_g} \log\left[D\left(\mathbf{f}_i^{s,g}\right)\right] - \frac{1}{n_t}\sum_{j=1}^{n_t} \log\left[1 - D\left(\mathbf{f}_j^t\right)\right] . \tag{4}$$

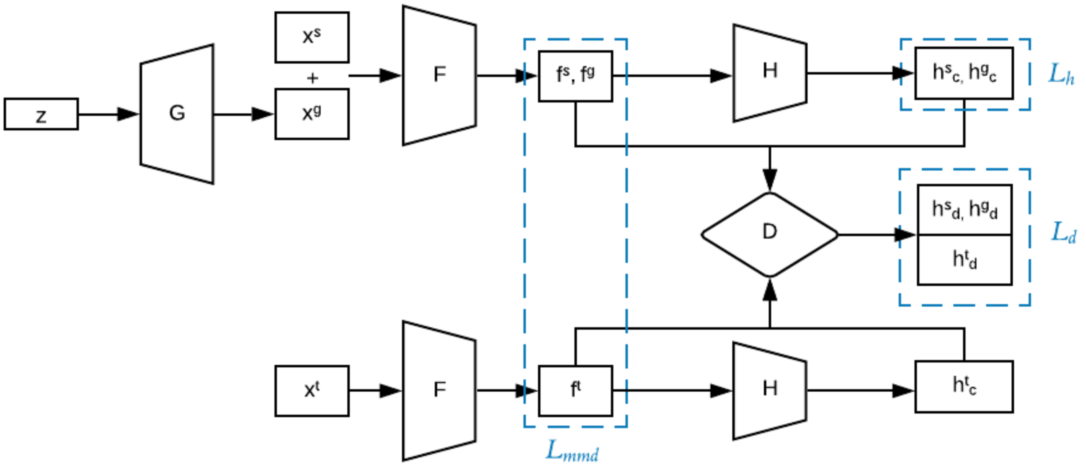


Figure 2: Proposed framework. We introduce a generator (G) that produces source samples from the unknown target classes. To ensure that these samples contain the correct information, we align the target feature distribution to the augmented source one via standard closed-set DA strategies, including an MMD-based loss and an adversarial domain classifier (D), with h_d the probability of classifying a sample in domain d. Furthermore, we encourage the generated samples to be classified as unknowns by the multi-class classifier (H). Our entire framework, including the generator, is trained in an end-to-end fashion.

Following CDAN [24], we modify this formulation to further condition the discriminator D on the prediction of the multi-class classifier H. Specifically, let $\mathbf{h}$ denote the multi-class probability vector output by the classifier H. We then write the discriminator loss as

$$L'_d = -\frac{1}{n_s+n_g}\sum_{i=1}^{n_s+n_g}\log\left[D\left(T_\otimes(\mathbf{f}_i^{s,g},\mathbf{h}_i^{s,g})\right)\right] - \frac{1}{n_t}\sum_{j=1}^{n_t}\log\left[1-D\left(T_\otimes(\mathbf{f}_j^{t},\mathbf{h}_j^{t})\right)\right] \ ,$$

where $T_\otimes(\cdot)$ is the multilinear map, i.e., outer product in our case, defined as $T_\otimes(\mathbf{f},\mathbf{h}) = \mathbf{f}\otimes\mathbf{h}$. This was shown by [24] to be more effective than concatenating $\mathbf{f}$ and $\mathbf{h}$.

Finally, as suggested by [24], to prevent the minimax problem from giving equal importance to the samples with uncertain predictions in the adaptation procedure, we reweight their influence according to uncertainty. Specifically, we measure uncertainty using the entropy $e(\mathbf{h}) = -\sum_{c=1}^{C}\mathbf{h}_c\log\mathbf{h}_c$, where $\mathbf{h}_c$ denotes the probability of classifying a sample in class c. This gives the discriminator loss

$$L_d = -\frac{1}{n_s+n_g}\sum_{i=1}^{n_s+n_g}e(\mathbf{h}_i^{s,g})\log\left[D\left(T_\otimes(\mathbf{f}_i^{s,g},\mathbf{h}_i^{s,g})\right)\right] - \frac{1}{n_t}\sum_{j=1}^{n_t}e(\mathbf{h}_j^{t})\log\left[1-D\left(T_\otimes(\mathbf{f}_j^{t},\mathbf{h}_j^{t})\right)\right] \ .$$

MMD-based distribution alignment. Another popular approach to align the source and target distributions in the closed-set DA literature consists of using the MMD [14]. This metric measures the discrepancy between two empirical distributions as the distance between their means in a reproducing kernel Hilbert space. In our context, we can express this as the loss function

$$L'_{mmd} = \left\| \frac{1}{n_s+n_g}\sum_{i=1}^{n_s+n_g}\phi\left(\mathbf{f}_i^{s,g}\right) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi\left(\mathbf{f}_j^{t}\right)\right\|_{\mathcal{H}}^2 \ , \tag{5}$$

where $\phi(\cdot)$ encodes the mapping to the reproducing kernel Hilbert space $\mathcal{H}$.

Following the same intuition as in the domain classifier case, we propose to re-weigh the contribution of each sample in this loss according to its uncertainty. This lets us re-write our MMD loss as

$$L_{mmd} = \left\| \frac{1}{n_s+n_g}\sum_{i=1}^{n_s+n_g}e(\mathbf{h}_i^{s,g})\phi\left(\mathbf{f}_i^{s,g}\right) - \frac{1}{n_t}\sum_{j=1}^{n_t}e(\mathbf{h}_j^{t})\phi\left(\mathbf{f}_j^{t}\right)\right\|_{\mathcal{H}}^2 \tag{6}$$

We then incorporate this loss function in (3) to obtain our complete learning formulation

$$\begin{aligned} \min_{\theta_G,\theta_F,\theta_H} \ & L_h(\theta_G,\theta_F,\theta_H) - \lambda_d L_d(\theta_G,\theta_F,\theta_D) \\ & + \lambda_m L_{mmd}(\theta_G,\theta_F,\theta_H) \\ \min_{\theta_D} \ & L_d(\theta_G,\theta_F,\theta_D) \ , \end{aligned} \tag{7}$$

where λ_m sets the relative influence of the MMD term. Note that, by setting either λ_d or λ_m to 0, our formalism allows us to employ a single distribution-alignment strategy. As will be evidenced by our experiments, our method remains highly effective in such cases. Note that, the generated data is unlikely to be from the known classes. For a dataset with $C-1$ classes, H is a C-way classifier, which we train by classifying the generated samples to class C. Generating samples from the other classes would confuse this classifier and thus degrade its accuracy.

4. Experiments

We compare our approach with the open-set domain adaptation methods **ATI-λ** [7], **OSBP** [35], **ROS** [6], **SE-CC** [31], **InheriTune** [20], **STA** [22], and **PGL** [27] on the three most challenging open-set DA datasets: *Office-Home*, *VisDA-17*, and *Syn2Real-O*. Furthermore, we report the results of two closed-set domain adaptation baselines representative of the two adaptation strategies we employ: the use of **MMD** [14] in a deep network, and the domain discriminator-based **DANN** [11]. Note that to make a fair comparison, and since pseudo-labeling can be incorporated in all existing open-set and closed-set DA methods to further improve their performance, we report the results of PGL [27] without pseudo-labeling in the tables. Finally, we also provide the results of not performing any domain adaptation using either a **ResNet-50** [16] or a **VGGNet** [37], according to the backbone used in the DA networks. For **MMD**, **DANN**, and **ResNet-50/VGGNet**, we utilize **OSVM** [18] to reject the unknown target samples.

All networks were trained using SGD with a learning rate of 0.001, a weight decay of 5×10^{-5}, and a momentum of 0.9. Following the learning rate annealing strategy of [11, 24], we adjust the learning rate by $(1+\alpha p)^{-\beta}$, where p is the training progress, and $\alpha = 0.001, \beta = 0.75$. For our approach, we used the same architectures as in [24] to define our classifier H and domain discriminator D. During training, we set λ_m to 1, and, relying on the progressive training strategy of [11, 24], increase λ_d from 0 to 1 as $\frac{1-\exp(-10p)}{1+\exp(-10p)}$, with p the training progress. We report the two widely-used metrics of normalized accuracy for the known classes (**OS***), and normalized accuracy for all classes (**OS**).

4.1. Datasets

Office-Home [40] is a challenging domain adaptation benchmark containing 15,500 images from 65 classes of everyday objects. There are 4 domains in the dataset: Art (**Ar**), Clipart (**Cp**), Product (**Pr**), and Real-World (**Rw**). For our experiments, we follow the same setting as in [22], consisting of taking the first 25 classes in alphabetical order as known classes and the remaining classes as unknown ones. For this set of experiments, all DA networks rely on a ResNet-50 [16] pre-trained on ImageNet as backbone network.

VisDA-17 [32] is a standard domain adaptation benchmark dataset comprising two domains, **Synthetic** and **Real**, which share 12 object classes. The **Synthetic** domain contains 152,397 synthetic images generated by 3D rendering. The **Real** domain consists of 55,388 real-world images taken from the MSCOCO [21] dataset. For our experiments, we follow the same protocol as in [35, 22], choosing 6 classes as the known set, and the remaining 6 classes as the unknown one. In this set of experiments, all DA networks employ a VGGNet [37] pre-trained on ImageNet as backbone network.

Syn2Real-O [33] constitutes the most challenging synthetic-to-real benchmark for open-set domain adaptation. It consists of synthetic and real objects from 12 categories which forms the known set in the **Synthetic** source domain and in the **Real** target domain. We take 50k MSCOCO images from irrelevant classes to form the unknown set in the target domain. Even though Syn2Real-O introduces 33 additional categories from ShapenetCore as unknowns in the source domain, we did not use that part of the data. This is consistent for all the methods we evaluate. In essence, we follow the open-set setting of [33], taking 12 classes as known ones for the source and target domains, and the other 69 COCO categories as the unknown classes in the target domain. In this set of experiments, all DA networks employ a ResNet-50 pre-trained on ImageNet as backbone network.

4.2. Results

As shown in Tables 1, 2, 3, and 4, our method outperforms the state-of-the-art baselines in most cases, consistently improving the average accuracy (OS), by 1.8%, 2% and 1.8% on *Office-Home*, *VisDA-17*, and *Syn2Real-O*, respectively. Note that *VisDA-17* and *Syn2Real-O* are among the most challenging open-set DA datasets.

Visualization: The t-SNE plot of Fig. 3(a) compares the distributions of the feature vectors f_i of the generated samples and the unknown target samples, computed after training the whole framework (including the feature extractor F). Note that our generated unknown samples (in orange) cover a large portion of the true unknown target sample distribution (in blue). While this confirms the effectiveness of our approach, small parts in the true distribution nonetheless remain unaccounted for, which, we believe, explains our slightly disappointing unknown class recognition accuracy. However, we expect this to be improved via the use of pseudo-labeling, which has proven to be effective in recent closed-set domain adaptation methods [8, 44, 43], and would thus easily extend to our formalism.

One potential source of errors in our approach would be that the generated samples depict known classes, instead of unknown ones. This, however, is prevented by classifier H, which forces the generated examples to be classified as unknown. Specifically, for a dataset with $C-1$ classes, H is a C-way classifier, which we train by clas-

Table 1: Normalized accuracy for the known classes (OS*), and for all classes (OS) (%) on the first 6 pairs of source/target domains the from *Office-Home* benchmark using ResNet-50 as backbone. **Ar**: Art, **Cp**: Clipart, **Pr**: Product, **Rw**: Real-World.

Method	**Ar→Cl**		**Ar→Pr**		**Ar→Rw**		**Cl→Rw**		**Cl→Pr**		**Cl→Ar**	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
ResNet+OSVM	37.5	38.7	42.2	42.6	49.2	51.4	53.8	55.5	48.5	50.0	39.2	40.3
DANN+OSVM	52.3	52.1	71.3	72.4	82.3	83.8	73.2	74.5	62.8	64.1	61.4	62.3
MMD+OSVM	50.6	52.4	65.5	67.9	77.8	80.7	57.8	60.1	62.9	65.4	**70.2**	**73**
ATI-λ [7]	53.1	54.2	68.6	70.4	77.3	78.1	74.3	75.3	66.7	68.3	57.8	59.1
OSBP [35]	56.1	57.2	75.8	77.8	83.0	85.4	75.5	77.2	69.2	71.3	64.6	65.9
STA [22]	58.1	-	71.6	-	**85.0**	-	75.8	-	69.3	-	63.4	-
ROS [6]	51.5	50.6	68.5	68.4	75.9	75.8	65.6	65.3	60.3	59.8	54.1	53.6
InheriTune [20]	60.1	-	70.9	-	83.2	-	75.7	-	70	-	64	-
PGL (w/o pro.) [27]	50.5	51.1	62.3	63.2	82.6	84.1	72.7	73.9	62.2	63.1	59.9	60.7
Ours	57.6	**58.6**	**79.3**	**80.5**	**85**	**86.5**	**76.4**	**77.6**	69.1	**71.7**	65.8	67.2

Table 2: Normalized accuracy for the known classes (OS*), and for all classes (OS) (%) on the remaining 6 source/target pairs from the *Office-Home* benchmark using ResNet-50 as backbone. **Ar**: Art, **Cp**: Clipart, **Pr**: Product, **Rw**: Real-World.

Method	**Pr→Ar**		**Pr→Cl**		**Pr→Rw**		**Rw→Ar**		**Rw→Cl**		**Rw→Pr**		**Avg.**	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
ResNet+OSVM	53.4	55.1	43.5	44.8	70.6	72.9	65.6	67.4	49.5	50.8	72.7	75.1	52.1	53.7
DANN+OSVM	63.5	64.5	46.0	46.3	77.2	78.3	70.5	71.3	55.5	56.2	79.1	80.7	66.2	67.2
MMD+OSVM	59.2	61.4	47.7	49.4	74.3	77.1	68.2	70.9	56.3	58.3	76.2	79.1	63.9	66.3
ATI-λ [7]	61.2	62.6	53.9	54.1	79.9	81.1	70.0	70.8	55.2	55.4	78.3	79.4	66.4	67.4
OSBP [35]	64.6	65.3	48.3	48.7	79.5	81.6	72.1	73.5	54.3	55.3	80.2	81.9	68.6	70.1
STA [22]	65.2	-	**53.1**	-	80.8	-	74.9	-	54.4	-	**81.9**	-	69.5	-
ROS [6]	57.6	57.3	47.5	46.5	71.1	70.8	67.1	67.0	52.3	51.5	72.3	72.0	62	61.6
InheriTune [20]	66.1	-	**54.2**	-	**81.3**	-	74.9	-	56.2	-	78.6	-	69.6	-
PGL (w/o pro.) [27]	58.9	59.7	44.6	44.9	75.2	76.5	72.2	73.3	49.9	50.6	76.3	77.7	63.9	64.9
Ours	**68.4**	**69.1**	**53.1**	**54.5**	**81.2**	**82.8**	**76.4**	**77.5**	**62.1**	**63.4**	81.8	**83.2**	**71.4**	**72.7**

sifying the generated samples to class C. To confirm that this approach is effective, we compare the distributions of the generated unknowns versus the known classes for the Syn2Real-O dataset in Fig. 3(b). Note that the generated unknown samples have only little overlap with the known classes. Moreover, despite the fact that the generated samples do not resemble images from the unknown classes, as shown in Fig. 4, the distribution of the resulting features extracted from these samples is close to that of real unknown samples, which makes them helpful to classify the target images.

To further analyse the effectiveness of the generated samples and how our method helps for feature alignment, we computed histograms of pairwise distances between the features of generated samples versus unknown target samples, known target samples, and source samples. As can be seen in Fig. 5, the generated images are more distant from the known source and target samples than from the unknown target examples.

4.3. Method Analysis

In this section, we evaluate different aspects of our approach.

Ablation Study: First, while our complete framework combines the MMD and a domain classifier to align the target and augmented source distributions, it can in principle rely on either one of these standard approaches individually. To evidence this, in Table 5(left), we compare our complete framework with these three alternatives, referred to as **Ours w** $L_h + L_{mmd}$, **Ours w** $L_h + L_d$, and **Ours w** $L_h + L_d$ **w/o Entropy Cond.**, and with the state-of-the-art PGL baseline. Note that, while accuracy is improved by

Table 3: Accuracy comparison on *VisDA-17* with VGGNet as backbone. OS* denotes normalized accuracy for the known classes, and OS denotes normalized accuracy for all classes.

Method	Bic	Bus	Car	Mot	Tra	Tru	unk	OS	OS*
VGGNet+OSVM	31.7	51.6	66.5	70.4	88.5	20.8	38	52.5	54.9
MMD+OSVM	39.0	50.1	64.2	79.9	86.6	16.3	44.8	54.4	56.0
DANN+OSVM	31.8	56.6	71.7	77.4	87.0	22.3	41.9	55.5	57.8
ATI-λ [7]	46.2	57.5	56.9	79.1	81.6	32.7	65.0	59.9	59.0
OSBP [35]	51.1	67.1	42.8	84.2	81.8	28.0	**85.1**	62.9	59.2
STA [22]	52.4	69.6	**59.9**	87.8	86.5	27.2	84.1	66.8	63.9
InheriTune [20]	53.5	69.2	62.2	85.7	85.4	32.5	**88.5**	68.1	64.7
PGL (w/o pro.) [27]	52.5	68.7	44.0	**91.6**	71.6	13.7	44.6	55.2	57.01
Ours	**66.2**	**83.1**	**59.9**	**88.4**	76.7	**41.2**	75.5	**70.1**	**69.2**

Table 4: Accuracy comparison on *Syn2Real-O* with ResNet-50 as backbone. OS* denotes normalized accuracy for the known classes, and OS denotes normalized accuracy for all classes.

Method	Aer	Bic	Bus	Car	Hor	Kni	Mot	Per	Pla	Ska	Tra	Tru	unk	OS	OS*
ResNet+OSVM	29.7	39.2	49.9	54.0	76.8	22.2	71.2	32.6	75.1	21.5	65.2	0.6	45.2	44.9	44.8
MMD+OSVM	51	56.9	55.2	45.2	77	27.1	61.8	**57.8**	44.7	35.1	73	9.6	14.3	46.8	49.5
DANN+OSVM	50.8	44.1	19.0	58.5	76.8	26.6	68.7	50.5	82.4	21.1	69.7	1.1	33.6	46.3	47.4
OSBP [35]	75.5	67.7	68.4	**66.2**	71.4	0.0	86.0	3.2	39.4	23.2	68.1	3.7	**79.3**	50.1	47.7
STA [22]	64.1	70.3	53.7	59.4	80.8	20.8	90.0	12.5	63.2	30.2	**78.2**	2.7	59.1	52.7	52.2
SE-CC [31]	82.1	**80.7**	59.7	50.0	80.6	**36.7**	83.1	56.2	56.6	21.9	57.7	4.0	70.6	56.9	55.8
PGL (w/o pro.) [27]	43.7	**80.7**	58.8	64.6	85.7	15.5	94.3	35.5	**87.9**	25	71.2	**12.6**	43.7	55.3	56.2
Ours	**86.3**	65.7	**69.7**	64.6	**88.7**	13.7	**91.4**	52	63.9	**34.6**	75.9	6.3	50.9	**58.7**	**59.4**

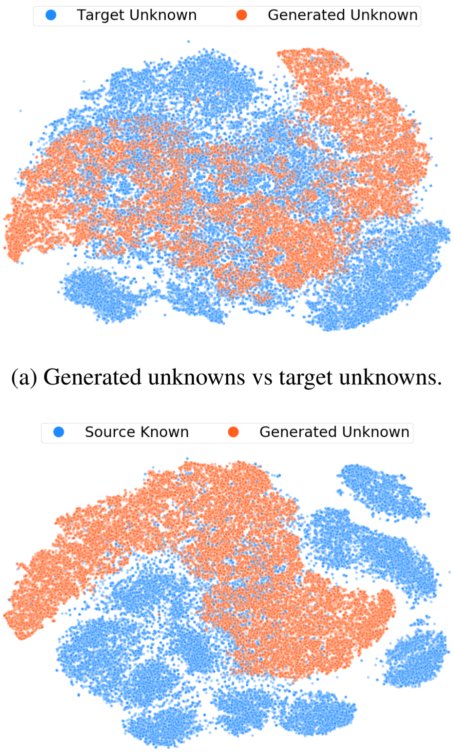
Table 5: Analysis of different aspects of our method on *Syn2Real-O*. (Left) Comparison of different distribution-alignment losses. (Right) Ablation study of the different components of our framework.

Method	OS	OS*
SE-CC [31]	56.9	55.8
Ours w $L_h + L_{mmd}$	55	59.2
Ours w $L_h + L_d$	54.8	55.7
Ours w $L_h + L_d$ w/o Entropy Cond.	54.2	54.7
Ours	**58.7**	**59.4**

Method	UNK	OS	OS*
Ours w Noise	20.9	44.8	46.8
Ours w/o Entropy Cond. in L_{mmd}	48.7	58	58.8
Ours w/o Entropy Cond. in L_d	43.9	57.7	58.9
Ours w/o Entropy Cond. in $L_d + L_{mmd}$	45.2	56.9	57.8
Ours	**50.9**	**58.7**	**59.4**

combining L_{mmd} and L_d, using each one separately within our model still consistently outperforms STA, and is comparable to SE-CC, thus showing the benefits and generality of our approach. Moreover, the accuracy of our approach with DAN only, as in OSBP, refereed to as **Ours w $L_h + L_d$ w/o Entropy Cond.** still outperform OSBP and the STA.

As a second analysis, we perform an ablation study to evaluate the influence of different components of our approach. In particular, to evidence the importance of generating samples that correspond to the unknown classes, as opposed to random noise treated as unknowns, we evaluate an **Ours w Noise** baseline, consisting of removing the generator from our approach and using random noise images instead. Furthermore, we report the results of our approach without the use of entropy conditioning to reweigh the samples in L_{mmd} and L_d, referred to as **Ours w/o Entropy Cond. in L_{mmd}**, **Ours w/o Entropy Cond. in L_d**, and **Ours w/o Entropy Cond. in $L_d + L_{mmd}$** respectively. As shown in Table 5(right), using random noise as unknown samples yields a huge performance degradation, showing



(a) Generated unknowns vs target unknowns.

(b) Generated unknowns vs source knowns.

Figure 3: t-SNE plots comparing the distributions of the generated unknowns with the target unknowns (a) and source knowns (b).

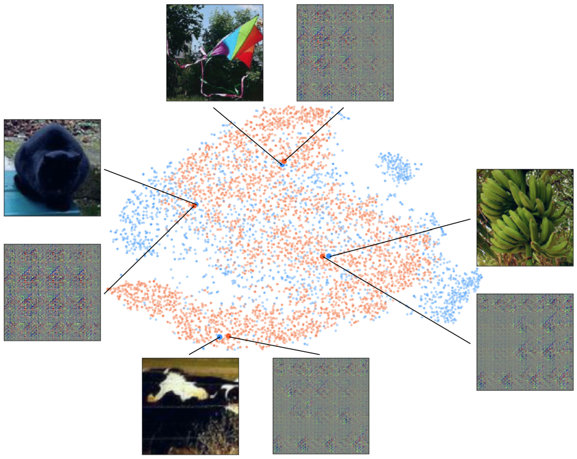


Figure 4: Target unknowns vs. generated images. While the generated images do not resemble target unknowns, their features are close, as evidenced by Figs. 3 and 5.

the importance of learning the distribution of the unknown data. By contrast, entropy conditioning only has little influence on the average accuracy. However, it helps to correctly classify the unknown samples.

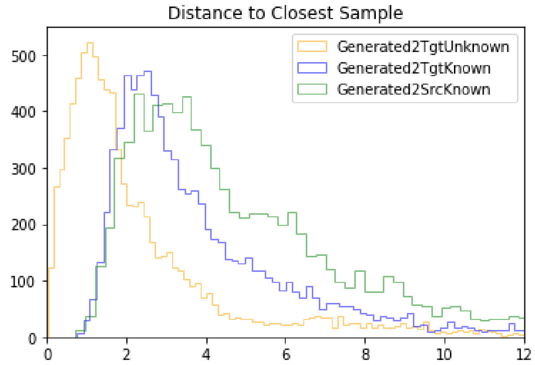


Figure 5: Histogram of pairwise distances between the generated samples and source/target samples.

Robustness Analysis to Varying Openness: Finally, we analyze the robustness of our approach to the openness of the data. To this end, following the same protocol as in [35, 22], we vary the openness of the Syn2Real-O data in $\{0.25, 0.5, 0.75, 0.9\}$ by removing different portions of the unknown samples. In Fig. 6, we compare the results of our approach with those of OSBP and STA. Our approach is more stable than OSBP and consistently outperforms both baselines by a large margin.

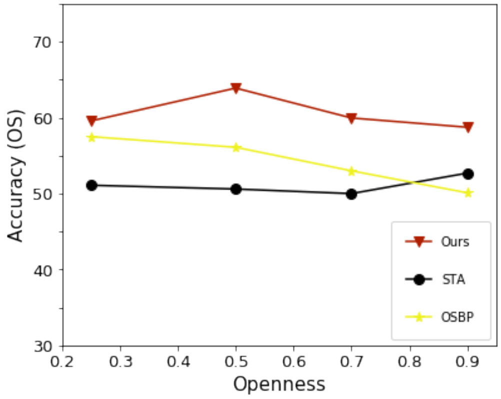


Figure 6: Accuracy vs. openness on Syn2Real-O.

5. Conclusion

We have introduced an approach to open-set DA that, in contrast to existing ones, does not aim to isolate the unknown target samples, but rather complements the source data by generating samples from the unknown target classes. In essence, this has allowed us to turn open-set DA into a closed-set problem, and thus to benefit from the great advances in closed-set DA. Our approach is simpler than existing open-set DA techniques, yet, as evidenced by our experiments on the three most challenging open-set DA benchmarks, consistently outperforms them. Furthermore, it is broadly applicable to most closed-set DA frameworks.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Mahsa Baktashmotlagh, Masoud Faraki, Tom Drummond, and Mathieu Salzmann. Learning factorized representations for open-set domain adaptation. In *Proc. Int. Conference on Learning Representations (ICLR)*, 2019.

[3] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1), 2010.

[5] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[6] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *European Conference on Computer Vision (ECCV)*. Springer, 2020.

[7] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proc. Int. Conference on Computer Vision (ICCV)*, 2017.

[8] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or distract: Exploit the margin of open set. In *Proc. Int. Conference on Computer Vision (ICCV)*, 2019.

[10] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for domain adaptation. *Proc. Int. Conference on Learning Representations (ICLR)*, 2018.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, pages 59:1–59:35, 2016.

[12] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 222–230, 2013.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[14] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample-problem. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2006.

[15] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] 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 (ECCV)*, 2014.

[18] Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. Multi-class open set recognition using probability of inclusion. In *Proc. European Conference on Computer Vision (ECCV)*, 2014.

[19] Pedro Ribeiro Mendes Júnior, Terrance E Boult, Jacques Wainer, and Anderson Rocha. Specialized support vector machines for open-set recognition. *arXiv preprint arXiv:1606.03802*, 2016.

[20] 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 conference on computer vision and pattern recognition (CVPR)*, 2020.

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, 2014.

[22] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proc. Int. Conference on Machine Learning (ICML)*, 2015.

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[25] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *Proc. Int. Conference on Computer Vision (ICCV)*, 2013.

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proc. Int. Conference on Machine Learning (ICML)*, 2017.

[27] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

[28] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. GCAN: graph convolutional adversarial network for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research (JMLR)*, 2001.

[30] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2010.

[31] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020.

[32] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *CoRR*, arXiv preprint arXiv:1710.06924, 2017.

[33] Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark for synthetic-to-real visual domain adaptation. *CoRR*, arXiv preprint arXiv:1806.09755, 2018.

[34] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proc. Int. Conference on Machine Learning (ICML)*, 2017.

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

[36] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2012.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conference on Learning Representations (ICLR)*, 2015.

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

[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.

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

[41] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.

[42] Guo-Sen Xie, Xu-Yao Zhang, Shuicheng Yan, and Cheng-Lin Liu. Hybrid cnn and dictionary-based models for scene recognition and domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(6), 2015.

[43] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *Proc. Int. Conference on Machine Learning (ICML)*, 2018.

[44] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.