

# Adversarial Network with Multiple Classifiers for Open Set Domain Adaptation

Tasnia Shermin, Guojun Lu, *Senior Member, IEEE*, Shyh Wei Teng, Manzur Murshed, *Senior Member, IEEE*, and Ferdous Sohel, *Senior Member, IEEE*

arXiv:2007.00384v3 [cs.CV] 7 Aug 2020

**Abstract**—Domain adaptation aims to transfer knowledge from a domain with adequate labeled samples to a domain with scarce labeled samples. Prior research has introduced various open set domain adaptation settings in the literature to extend the applications of domain adaptation methods in real-world scenarios. This paper focuses on the type of open set domain adaptation setting where the target domain has both private ('unknown classes') label space and the shared ('known classes') label space. However, the source domain only has the 'known classes' label space. Prevalent distribution-matching domain adaptation methods are inadequate in such a setting that demands adaptation from a smaller source domain to a larger and diverse target domain with more classes. For addressing this specific open set domain adaptation setting, prior research introduces a domain adversarial model that uses a fixed threshold for distinguishing known from unknown target samples and lacks at handling negative transfers. We extend their adversarial model and propose a novel adversarial domain adaptation model with multiple auxiliary classifiers. The proposed multi-classifier structure introduces a weighting module that evaluates distinctive domain characteristics for assigning the target samples with weights which are more representative to whether they are likely to belong to the known and unknown classes to encourage positive transfers during adversarial training and simultaneously reduces the domain gap between the shared classes of the source and target domains. A thorough experimental investigation shows that our proposed method outperforms existing domain adaptation methods on a number of domain adaptation datasets.

**Index Terms**—Open set domain adaptation, adversarial domain networks, multi-classifier based weighting module.

## I. INTRODUCTION

DEEP learning models for computer vision tasks usually require a massive amount of labeled data entailing highly laborious work for annotating data [1], [2], [3], [4]. An alternative is to use labeled data from a related (source) domain to boost the performance of the model in a target domain. However, as the source and target data may have domain gaps such as different illumination set-ups, and perspectives, synthesized data by using different variants of sensors, the performance of this approach may suffer. Existing domain adaptation (DA) methods aim to decrease the above-mentioned domain divergences either by using distribution matching methods [5], [6], [7], [8] or by transforming samples from one domain to another through generative models [9], [5], [10], [11], [12], [13], [14].

Tasnia Shermin, Guojun Lu, Shyh Wei Teng, and Manzur Murshed are with the School of Science, Engineering and Information Technology, Federation University, Churchill-3842, Australia (email: {t.shermin, guojun.lu, shyh.wei.teng, manzur.murshed}@federation.edu.au)

Ferdous Sohel is with Murdoch University, WA-6150, Australia (email: f.sohel@murdoch.edu.au)

Generally, it is assumed that label sets across the source and target domains are identical (closed set domain adaptation [15], [16], [17]), as shown in Fig. 1a. However, such a simplified setting only has limited real-world applications. Open set domain adaptation [18], [19], and partial domain adaptation [20], [21], [22] methods have been proposed to ease the closed set domain adaptation assumption. Open set and partial domain adaptation settings assume source or target domain private label sets besides the identical (shared) label sets. The domain adaptation models-based on these domain adaptation settings are required to recognize the samples of the target domain private label sets as 'unknown' class and the samples belonging to shared label sets as known classes.

As illustrated in Fig. 1b, partial domain adaptation [20], [21], [22] setting assumes that the source domain label space is a superspace of the target domain label space. Open set domain adaptation setting proposed by Bustos et al. [19] requires images from unknown classes both in the source and target domain besides the shared known classes. However, this is not a cost-effective setting for open set domain adaptation as it requires a collection of a large number of unknown source samples with no prior knowledge about target labels. During training, Bustos et al. [19] bound the domain adaptation model to align unknown classes of the target domain towards unknown classes of the source domain. This may enforce a firm boundary for the unknown classes. During testing, samples from unknown classes other than the trained unknown classes may confuse the model.

Open set domain adaptation by back-propagation (OSBP) [18] setting takes another step towards the practical domain adaptation scenario by removing unknown classes from the source domain such that the source label space is a subset of the target label space (Fig. 1d). This means that the OSBP domain adaptation setting has samples from unknown classes only in the target domain and encourages the domain adaptation model to learn to detect an unknown target sample as unknown when it does not belong to the known classes. Thus, this setting assists in training the domain adaptation model with a broader unknown class boundary and detect unknown samples during testing better than the previous setting [19]. This paper focuses on improving the performance of domain adaptation model for the OSBP domain adaptation setting, which seems to be realistic and challenging as the source domain has less number of classes compared to the target domain.

Prevalent distribution matching domain adaptation methods [5], [6] cannot be applied to the OSBP domain adaptation set-

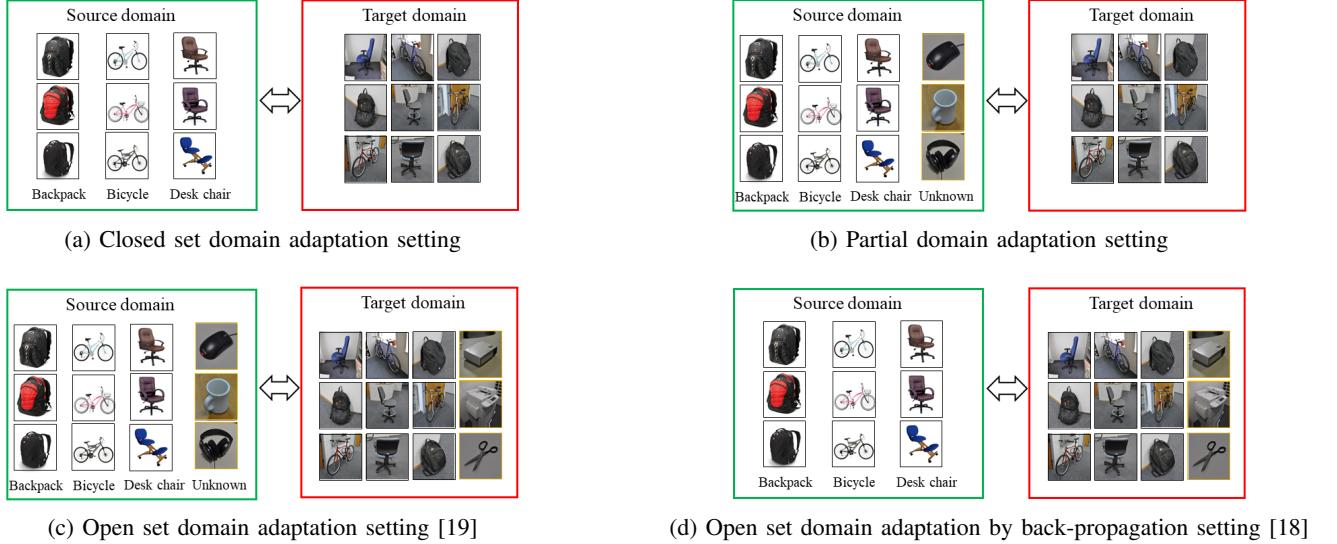


Figure 1: (a) The closed set domain adaptation setting assumes that both source and target domains consist of images only of the same set of classes. (b) Partial domain adaptation setting assumes that the source domain label space (classes) is the superspace of the target domain label space. This adaptation setting includes images of unknown classes in the source domain only. (c) Open set domain adaptation setting proposed by Bustos et al. [19] requires images of unknown classes in both source and target domains, i.e., both source and target domains contain images that do not belong to the label space of interest. (d) Open set domain adaptation setting introduced by Saito et al. [18] requires images of unknown classes only in the target domain besides the classes of interest. We focus on the domain adaptation setting illustrated in (d).

ting as the absence of unknown samples in the source domain does not allow the unknown samples of the target domain to be aligned. Saito et al. [18] have proposed a generative model to address the OSBP domain adaptation setting where they enable the classifier to draw a rough boundary between the source and target samples (i.e., initially all the target samples will be classified as ‘unknown’) and the generator has to separate the target samples into known and unknown classes adversarially. The adversarial learning between the generator and the classifier depends on the pseudo decision of the classifier. However, their proposed method does not explore any underlying domain discriminative information to assess the pseudo decision of the classifier before the adversarial training. Also, they set an empirical fixed threshold for generator-classifier adversarial training to differentiate known target samples from unknown ones. We argue that this concept of relying only on the rough decision of the classifier may encourage the negative transfer of target samples.

Pan et al. [23] stated that a DA model is prone to negative transfers when it lags behind a non-DA model (which is trained only on the source domain) in performance. The fixed threshold (0.5) for constructing boundary between known and unknown target samples leads to biased adversarial learning. That is, for a target sample, when the classifier assigns an unknown probability low than the threshold based on its pseudo decision boundary, the generator will always be encouraged to align that sample towards known classes even if they belong to unknown classes. The OSBP method struggles to perform better than non-DA classifier for some tasks discussed in section IV-C because of negative transfers by aligning unknown target samples towards known classes.

To this end, we propose to extend their domain adversarial network by integrating a new multi-classifier based weighting module to the network. We refer to the extended generative model as the multi-classifier based adversarial domain adaptation model. To reduce negative transfers, first, we evaluate the underlying discriminative domain information of the known and unknown target samples, and then assign them with distinguishable weights which are more representative to their similarities to the source domain. In particular, the underlying domain information of target samples is measured by evaluating discriminative label information based on the resemblance to each source domain classes and probable similarity with the source domain classes when measured against the probability of belonging to the unknown classes of the target domain. This ensures that the proposed weighting module provides a rough estimate of the underlying domain of the target samples. Then based on our generated weights, the generative module performs adversarial domain adaptation and aligns known target samples towards known source samples and rejects the unknown target samples. Thus, in the proposed DA model, the generative module is not forced to draw a threshold driven boundary between known and unknown target samples, which may initiate negative transfers as the baseline method. The proposed model improves performance over the previous method [18], which indicates it constructs a good boundary between known and unknown target samples by eliminating negative transfers.

Several attempts to address partial DA [22], [20], universal DA [24], and open set DA [25] by generating weights are identified in the literature. Zhang et al. [22] utilize the output of an auxiliary domain discriminator to derive the probability of

the source samples belonging to the target domain and assign weights to the source samples accordingly for partial DA. For improving prior partial DA methods participating in negative transfer [21], [26], Example Transfer Network (ETN) [20] degrades the weight of source images from source private classes before integrating to the source classifier and places a discriminative domain classifier to quantify sample transferability. To generate weights for source samples belonging to the shared label sets, Universal Adaptation Network (UAN) [24] integrates domain similarity and prediction uncertainty. Unlike our proposed method, UAN does not integrate underlying label information to evaluate domain similarity between the source and target samples. Above-mentioned methods concentrate on assigning weights to the source samples, whereas the OSBP DA setting demands to assign weights to the unlabeled target samples. This is more challenging than weighting source samples as we do not know the labels of target samples and the shared label space during training. We propose a new multi-classifier based weighting scheme in an adversarial DA method for the OSBP DA setting. Separate to Adapt (STA) [25] method assigns weights to a target sample based on its highest similarity to one of the source domain classes. On the contrary, our proposed weighting module assesses domain information based on a target samples' similarity to known classes and dissimilarity to the unknown class, and then assign identifiable weights (please refer to Section III-C2 for details). The main contributions of this paper are as follows:

- We propose a domain adversarial model by integrating a new multi-classifier module in the OSBP domain adversarial model [18] for the open set domain adaptation setting that has access to unknown classes only in the target domain. The multi-classifier structure introduces a weighting scheme in the proposed model, which assesses fundamental domain information based on distinctive label information for assigning identifiable weights to the known and unknown target samples. This enhances the positive transfer of target samples and facilitate the adversarial training. Unlike the previous method [18], which requires an assumption of the known-unknown boundary threshold, the proposed method is capable of automatically discovering the boundary between known and unknown target samples.
- We conduct comprehensive experiments and demonstrate that our proposed model reduces the rate of negative transfer and achieves better performance than contemporary domain adaptation methods on several datasets.

The proposed method can be applied to many multimedia applications such as image or video classification. Section II presents a brief discussion about contemporary DA methods. Details on our proposed method are described in Section III. Thorough performance comparison and analysis of the proposed method with contemporary DA approaches on different datasets are provided in Section IV.

## II. RELATED WORK

In this section, we briefly review recent domain adaptation methods based on the inter-domain relationship of label set

constraints. These methods include those from closed set domain adaptation, partial domain adaptation, open set domain adaptation, and universal domain adaptation.

### A. Closed Set Domain Adaptation

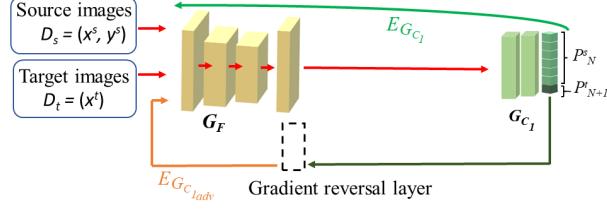
Closed set DA methods concentrate on reducing the divergence between the source and the target domains. Recent works have shown that because of its setting, closed set DA methods can exploit domain invariant features by explicitly reducing the domain divergence upon the supervised deep neural network structures. The development of deep learning based closed set DA methods [27], [13], [16], [15], [10], [28], [29], [17], [30] is originated from prior shallow DA methods [31], [32], [33], [34], [35], [36]. Closed set DA methods fall into three main categories. The first category of methods is based on static moment matching, such as Maximum Mean Discrepancy (MMD) [37], [15], [16], [28], Central Moment Discrepancy (CMD) [38], and second-order statistics matching [39]. The second category of methods adapts the adversarial loss concept of GAN [40] and initiates the generation of images that are non-discriminative to the shared label space of source and target domain [30], [6], [10]. Furthermore, domain adversarial methods align pixels and features from both domains and synthesize labeled target images for data augmentation [41], [42], [43], [44], [45], [5]. Saito et al. [13] utilize the probabilistic outputs of two classifiers' to measure domain discrepancy loss and update both the classifiers based on this loss adversarially. However, we use two different types of classifiers and a discriminator for measuring underlying domain similarity and utilize this similarity to compute final domain discrepancy loss by updating only the main classifier. Recent research has explored Cycle-Consistent GAN [46] for developing CycleGAN-based [47], [48], [49] DA methods. The final category of methods leverages Batch Normalization statistics for adapting domains to a canonical cone [50], [51].

### B. Partial Domain Adaptation

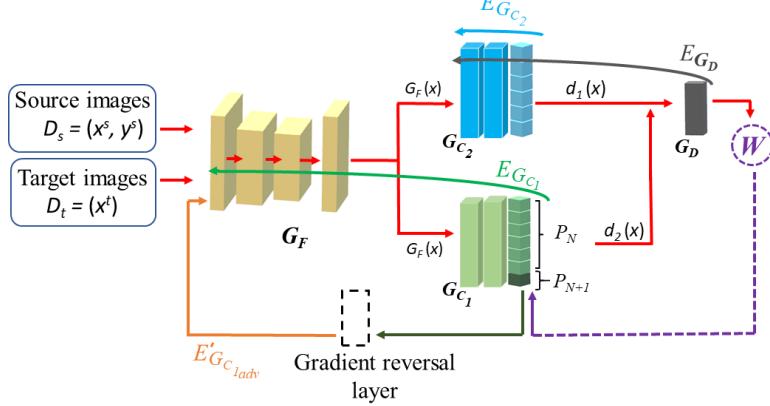
The vanilla closed set DA's assumption that the source and target domains share the same label space does not hold in partial DA. The setting of partial DA assumes a target domain that has a fewer number of classes than the source domain. Cao et al. [21] use multiple domain discriminators along with class-level and instance-level weighting mechanism to obtain class-wise adversarial distribution matching for solving partial DA. Cao et al. [26] refine Selective Adversarial Network (SAN) [21] by using only one adversarial network and integrating the class-level weight to the source classifier.

### C. Open Set Domain Adaptation

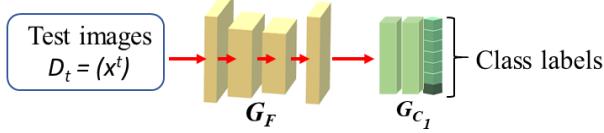
Open set DA methods aim to reject outliers or images from unknown classes while correctly recognizing inliers or images from the classes of interest (shared label space). Multi-class open set SVM [52] is designed to reject images from unknown classes. In this method, the SVMs are trained to assign probabilistic decision scores to samples and reject unknown samples by a threshold. Bendale et al. [53] integrates



(a) Block diagram of the OSBP domain adaptation model [18].



(b) Training phase of the proposed domain adaptation model.



(c) Testing phase of the proposed domain adaptation model.

Figure 2: (a) Block diagram of the baseline method [18]. (b) Block diagram of the training phase of our proposed adversarial domain adaptation network with a multi-classifier based weighting scheme. Here,  $G_F$ ,  $G_{C_1}$ ,  $G_{C_2}$  and  $G_D$  denotes the generator, domain classifier, non-adversarial supplementary source and domain classifier respectively.  $E_{G_{C_1}}$  and  $E_{G_{C_1,adv}}$ ,  $E_{G_{C_2}}$ , and  $E_D$  are errors for optimizing  $G_F$  and  $G_{C_1}$ ,  $G_{C_2}$ , and  $G_D$  respectively.  $G_F(x)$  stands for the generated features by the generator, which is fed into the domain classifier and supplementary source classifier simultaneously.  $W$  is the generated weights for target samples. (c) A pictorial illustration of the proposed model during the testing phase. Red arrows denote forward passes while other arrows represent a backward pass.

an OpenMax layer upon deep neural networks for exploiting them in open set recognition. The OpenMax layer assists the network in estimating the probability of an image coming from an unknown class. To generate unknown samples for open set recognition, Ge et al. [54] combines a generative model with the OpenMax layer, and to reject unknown samples during testing, this method defines a threshold.

As shown in Fig. 1c, Bustos et al. [19] considers images from outliers (out of shared label space) of both source and target domains as ‘unknown’ class. This method aligns target samples to source samples by an Assign-and-Transform-Iteratively (ATI) and trains SVMs for classification. This open set DA method takes advantage of the target samples from unknown classes to align with source samples from unknown classes. Saito et al. [18] modified the open set DA setting for addressing more practical DA cases. Their modified setting does not require unknown classes in the source domain. They address this DA by adversarially training a classifier with an extra category named ‘unknown’. Unlike our method, OSBP does not explore any domain information instead enforces hard threshold and STA [25] depends only on the decision

of source-only trained classifier for separating unknown target samples, which can give uncertain predictions for target samples. Our proposed method overcomes such shortcomings and improves classification performance over their methods.

#### D. Universal Domain Adaptation

Recent research introduces a universal DA setting that imposes no previous knowledge on the source, and target domain label sets [24]. This is another valuable step towards addressing a practical adaptation scenario. The authors proposed a Universal Adaptation Network (UAN), which combines domain similarity and prediction uncertainty while generating sample weights for finding shared label sets.

### III. PROPOSED METHODOLOGY

In this section, we formally describe the open set domain adaptation setting, which is the focus of this work, discuss the limitations of the baseline method and present our proposed method.

### A. Domain Adaptation Setting

The open set domain adaptation setting of our focus constitutes a source domain  $D_s = (x_i^s, y_i^s)_{i=1}^{n_s}$  of  $n_s$  labeled instances associated with  $|C_s|$  classes, which are drawn from distribution  $p_s$  and a target domain  $D_t = (x_j^t)_{j=1}^{n_t}$  of  $n_t$  unlabeled instances drawn from distribution  $p_t$ , where  $p_s \neq p_t$ . We denote the class labels of the target and source domains as  $C_t$  and  $C_s$  respectively. The shared label space is denoted as  $C = C_s \cap C_t$ .  $\bar{C}_t = C_t \setminus C$  represents the label sets private to the target domain, which should be recognized as ‘unknown’. In this type of open set domain adaptation, it is difficult to identify which part of the target label space  $C_t$  is shared with the source label space  $C_s$  because the target domain is fully unlabeled and  $C_t$  is unknown at the training time. It is challenging to differentiate between known and unknown target samples as we do not have any trace of the target sample labels.

### B. Baseline Domain Adversarial Model

In this section, we discuss in brief the OSBP [18] model and its tendency to initiate negative transfers.

The OSBP method (Fig. 2a) is an adversarial DA model that aims to reduce divergence between the source and target domains by learning transferable features in a two-player minimax game in line with existing domain adversarial networks [29], [30]. The first player of the model is a domain classifier  $G_{C_1}$  and the second player is a feature generator  $G_F$ . The final objective of the OSBP method is to correctly classify known target samples as corresponding known class and target samples belonging to unknown classes as ‘unknown’.

The feature generator  $G_F$  takes inputs from both source domain  $D_s$  and target domain  $D_t$  at the same time. The domain classifier  $G_{C_1}$  takes features from  $G_F$  and outputs  $N+1$  dimensional probability, where  $N$  specifies the number of known or source categories ( $C_s$ ) and the probability for the unknown category is indicated by the  $(N+1)^{th}$  index. During the forward pass, within  $G_{C_1}$ , the features are transformed to a  $N+1$ -dimensional class probability through softmax function as,  $\sigma(z) = \exp(z)/(\sum_{i=1}^{N+1} \exp(z_i))$ , where  $z$  is the logit vector.

The OSBP method intends to construct a pseudo decision boundary for unknown classes. As the target domain is unlabeled during training, the domain classifier  $G_{C_1}$  is weakly trained to construct a pseudo decision boundary between known source samples and target samples by putting the target samples on the side of the unknown category. The OSBP method trains the domain classifier  $G_{C_1}$  to output  $P(y = N+1|x_j^t) = T$  for ‘unknown class’. Then the feature generator  $G_F$  is trained to deceive the domain classifier  $G_{C_1}$  adversarially. The feature generator  $G_F$  is trained with the ability to increase or decrease the ‘unknown’ class probability  $P(y = N+1|x_j^t)$  of the classifier  $G_{C_1}$  for maximizing the error of  $G_{C_1}$  and align target samples to known or unknown classes. Also, the OSBP method assumes that the empirical threshold value  $T = 0.5$  dictates the generative model to construct a good boundary between known and unknown target samples.

The OSBP method trains the domain classifier and the generator on source samples first as follows,

$$E_{G_{C_1}} = \frac{1}{n_s} \sum_{i=1}^{n_s} L_{G_{C_1}}(G_F(x_i^s), y_i^s). \quad (1)$$

Here,  $L_{G_{C_1}}$  is the standard cross-entropy loss function for minimizing the error of  $G_{C_1}$ . Then a binary cross-entropy loss is used for maximizing the error of  $G_{C_1}$  adversarially to separate known and unknown target samples as follows,

$$\begin{aligned} E_{G_{C_1\text{adv}}} &= -\frac{1}{n_t} \sum_{j=1}^{n_t} T \log(P(y = N+1|x_j^t)) \\ &\quad - \frac{1}{n_t} \sum_{j=1}^{n_t} (1-T) \log(1 - P(y = N+1|x_j^t)). \end{aligned} \quad (2)$$

The overall training objective of the OSBP method is,

$$\begin{aligned} \theta_{G_{C_1}} &= \underset{\theta_{G_{C_1}}}{\operatorname{argmin}} E_{G_{C_1}} + E_{G_{C_1\text{adv}}} \\ \theta_{G_F} &= \underset{\theta_{G_F}}{\operatorname{argmin}} E_{G_{C_1}} - E_{G_{C_1\text{adv}}}. \end{aligned} \quad (3)$$

The training objective indicates that the domain classifier  $G_{C_1}$  tries to set ‘unknown’ class probability  $P(y = N+1|x_j^t)$  equal to  $T$ , on the other hand the generator  $G_F$  tries to make  $P(y = N+1|x_j^t)$  different from  $T$  for maximizing the value of  $E_{G_{C_1\text{adv}}}$ . For calculating the gradient of  $E_{G_{C_1\text{adv}}}$  efficiently, the OSBP method utilizes a gradient reversal layer proposed by [6].

**Negative Transfer in the OSBP Method:** The OSBP method does not evaluate any underlying domain level discerning characteristics but relies only on the pseudo decision of  $G_{C_1}$ . The lack of such domain knowledge and enforcement of an assumed boundary threshold  $T = 0.5$  value for separating known and unknown classes will harm the model as the generator will attempt to align all the target samples with  $P(y = N+1|x_j^t) < 0.5$  to known classes during training. As a result, the model will be deprived of the opportunity to learn such image features which should be recognized as unknown even when  $P(y = N+1|x_j^t)$  is less than 0.5. For example, during training, the domain classifier  $G_{C_1}$  assigns  $P(y = N+1|x_j^t) = 0.4$  for a target sample and the rest 0.6 of the softmax probability is distributed over other  $1, 2, \dots, N$  indices of  $G_{C_1}$  with no index holding  $P \geq 0.4$ . This probability outcome distribution suggests the sample should be aligned towards the unknown class. However, the generator will always find it easier to decrease  $P(y = N+1|x_j^t) < 0.4$  to maximize the error of  $G_{C_1}$  and align it towards known classes as it does not explore underlying domain knowledge before adversarial training. Thus, the model will be exposed to negative transfers during training and testing i.e., target samples from unknown classes will be aligned to known classes.

To reduce the propensity for such negative transfers in the OSBP method and improve classification performance, we propose to extend their domain adversarial model. Our proposed multi-classifier based domain adversarial model is discussed in the next section.

### C. Proposed Multi-classifier Based Domain Adversarial Model

The limitations of the OSBP method is mainly due to the lack of an indicator of the likelihood of a target sample belonging to known or unknown classes. This prompted us to design a method to produce the indicator of target samples belonging to known or unknown classes beforehand the adversarial training to facilitate DA. The proposed method is illustrated in Fig. 2b. To investigate underlying domain information of target samples for reducing negative transfers, we propose to integrate a multi-classifier based weighting module in the baseline network. Our proposed method comprises of two modules: 1)Adversarial module and 2) Multi-classifier based weighting module.

*1) Adversarial module:* The adversarial module of our proposed method has a similar structure to the OSBP method. The first player of the model is a domain classifier  $G_{C_1}$  which is trained to distinguish the features of the source domain from the target domain. The second player is a feature generator  $G_F$  which is simultaneously trained to reduce feature distribution divergence in the opposite direction of the domain classifier. However, our proposed method follows a different adversarial optimization procedure for distinguishing unknown target samples from the known target samples compared to the OSBP. The ultimate goal is to train a source domain classifier that is transferred to the target domain classifier with an extra category named ‘unknown’.

The proposed weighting module quantifies each target sample with a generated weight  $W(x_j^t)$  (Section III-C2), which is an encoding of the underlying discriminative domain information. In particular, prior to adversarial DA, we assign identifiable weights  $W(x_j^t)$  to known and unknown target samples based on their similarity to source domain to facilitate the generator in deciding whether to decrease or increase the ‘unknown’ class probability  $P(y = N+1|x_j^t)$  for maximizing the error of  $G_{C_1}$  and eventually align the target samples to known classes or ‘unknown’ class. Therefore, the generator does not have to draw an empirical threshold driven boundary between known-unknown target samples by depending only on the pseudo decision of the classifier, which may encourage negative transfer.

We use the standard cross-entropy loss optimization, as illustrated in Equation (1) for minimizing the error of the domain classifier  $G_{C_1}$ . To distinguish known from unknown target samples and simultaneously maximize the error of the domain classifier  $G_{C_1}$  adversarially, we use a binary cross-entropy loss. We infuse our computed weight measure for the target samples in the optimization procedure as follows,

$$\begin{aligned} E'_{G_{C_1 \text{adv}}} &= -\frac{1}{n_t} \sum_{j=1}^{n_t} W(x_j^t) \log(P(y = N+1|x_j^t)) \\ &\quad - \frac{1}{n_t} \sum_{j=1}^{n_t} (1 - W(x_j^t)) \log(1 - P(y = N+1|x_j^t)). \end{aligned} \quad (4)$$

The mini-max game between the generator and domain classifier in the adversarial model is equivalent to aligning target samples towards known classes of the source domain or ‘unknown’ class based on their weights.

*2) Multi-classifier based weighting module:* In this section, we present the conceptual details of the weighting scheme of our proposed DA method.

**Overview:** The main challenge in our proposed method, as illustrated in Equations (1) and (4), is the way of measuring the probability of each target samples resembling the source domain on the basis of which the boundary between known and unknown classes is to be constructed. We aim to develop a weight measure  $W(x_j^t)$  for each target samples based on their discriminative label and domain information. We propose to integrate a supplementary source classifier  $G_{C_2}$  in the adversarial model (Fig. 2b) to determine the similarity of target samples to individual known-source labels  $C_s$ . The combined similarity to each known classes represents the similarity to the source domain. And, utilize the pseudo-decision of  $G_{C_1}$  to compute the combined similarity to all known classes (i.e., similarity to source domain) measured against target private labels  $\bar{C}_t$ . This ensures the exploitation of probable underlying label information of target samples concerning both known and unknown classes.

We further introduce a non-adversarial supplementary domain classifier  $G_D$  in the model to evaluate underlying domain information and produce a weight for target samples. The non-adversarial supplementary domain classifier  $G_D$  assumes that the target samples belonging to the shared label space  $C$  are closer to the source domain samples than  $\bar{C}_t$ . Being inspired by a prior work [55] that integrated the label information into the domain discriminator, we combine the domain similarity measure of  $G_{C_1}$  and  $G_{C_2}$ , and encode in  $G_D$ . Now that the supplementary domain classifier  $G_D$  holds the encoded information required to serve our purpose,  $G_D$  is jointly trained with  $G_{C_2}$  to distinguish the source domain samples from the target domain samples by utilizing the Sigmoid probability of classifying each target sample ( $x_j^t$ ) to the source domain. The output ( $W(x_j^t)$ ) of  $G_D$  for target samples gives the probability of a sample belonging to the shared label space.

This constitutes our multi-classifier based weighting module. The purpose of the weighting scheme is to assign either high or low weights to the target samples depending on their similarity to the source domain and reduce the chances of negative transfer. Generated weights are back-propagated to the adversarial module for optimizing the generator  $G_F$  and the domain classifier  $G_{C_1}$  to construct boundary between known and unknown classes.

**Mechanism in detail:** We place the supplementary source classifier  $G_{C_2}$  to predict the source class labels with a leaky-softmax function [20], which maintains the total probability of less than 1. The supplementary source classifier  $G_{C_2}$  converts features of the generator  $G_F$  to  $|C_s|$ -dimensional class probabilities as follows,

$$\bar{\sigma}(l) = \frac{\exp(l)}{|C_s| + \sum_{c=1}^{|C_s|} \exp(l_c)} \quad (5)$$

where  $l$  is the logit vector. The parameters of  $G_{C_2}$  is trained only on the source samples; therefore, unlike source samples, target samples will have smaller logits or uncertain predictions. We define the probability of each sample belonging to the

source domain based on known-source label information  $d_1(x)$  as follows,

$$d_1(x) = \sum_{k=1}^{|C_s|} G_{C_2}^k(G_F(x)) \quad (6)$$

where,  $G_{C_2}^k(G_F(x))$  is the probability of a sample belonging to the  $k^{th}$  known class. The element-sum ( $d_1(x)$ ) of the leaky-softmax outputs for samples resembling the source domain will be high or close to 1 whereas, samples dissimilar to source domain will yield low or close to 0 outputs. That is, the higher the value of  $d_1(x_j^t)$  is, the higher the chance that a target sample lies in the vicinity of shared label space  $C$ . On the other hand, the smaller the value of  $d_1(x_j^t)$  is, the more probable that the target sample comes from  $\overline{C}_t$ . We train  $G_{C_2}$  by a multiclass one-vs-rest binary loss for the  $|C_s|$ -class classification as,

$$\begin{aligned} E_{G_{C_2}} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{k=1}^{|C_s|} & y_{i,k}^s \log G_{C_2}^k(G_F(x_i^s)) \\ & + (1 - y_{i,k}^s) \log(1 - G_{C_2}^k(G_F(x_i^s))) \end{aligned} \quad (7)$$

where  $y_{i,k}^s$  denotes the ground-truth label for source example  $x_i^s$  and the probability of each sample  $x$  belonging to class  $k$  is  $G_{C_2}^k(G_F(x))$ . This similarity measure defined so far is still exposed to risk as to the value of  $d_1(x)$  for target samples can be uncertain. To further support the similarity measure, we compute the probability of a sample belonging to  $C$  when measured against the ‘unknown’ class probability from the domain classifier  $G_{C_1}$  as follows,

$$d_2(x) = (1 - P(y = N + 1|x)). \quad (8)$$

Target samples residing in the shared label set are likely to produce higher  $d_2(x)$  than unknown target samples. Now, we define our final similarity measure supported by  $G_{C_1}$  and  $G_{C_2}$  as,

$$G_D(G_F(x)) = (d_2(x))(d_1(x)). \quad (9)$$

Now,  $G_D(G_F(x_j^t))$  can be seen as the complete measure of the likelihood of target samples belonging to shared label space  $C$  i.e., for target samples, the higher the value of  $G_D(G_F(x))$  is the more probable that it belongs to the shared classes. For the convenience of understanding, we represent the possible cases of outcomes from the two deciding factors (i.e.,  $d_1(x_j^t)$  and  $(d_2(x_j^t))$ ) for computing the similarity measure of the known and unknown target samples as follows:

**A<sub>H</sub>:** For the target samples belonging to the shared label space  $C$ , it is highly likely the output of  $d_1(x_j^t)$  will be high.

**A<sub>L</sub>:** For the target samples belonging to the target private label space  $\overline{C}_t$ , it is highly likely the output of  $d_1(x_j^t)$  will be low.

**B<sub>H</sub>:** For the target samples belonging to the shared label space  $C$ , it is highly likely the output of  $d_2(x_j^t)$  will be high.

**B<sub>L</sub>:** For the target samples belonging to the target private label space  $\overline{C}_t$ , it is highly likely the output of  $d_2(x_j^t)$  will be low. Note that, here, high means close to 1 and low means close to 0. In the cases below,  $(A, B)$  denotes the occurrence of  $A$  and  $B$  for measuring the similarity of a target sample to the source domain.

**Case 1 (A<sub>L</sub>, B<sub>L</sub>):** For this case, the computed similarity measure (Equation (9)) will be low and this will assist the generator in deciding to increase the value of  $P(y = N + 1|x_j^t)$  for maximizing the error of the domain classifier  $G_{C_1}$  and align the sample towards the ‘unknown’ class.

**Case 2 (A<sub>L</sub>, B<sub>H</sub>):** For this case, the computed similarity measure (Equation (9)) will be low and this will assist the generator in deciding to increase the value of  $P(y = N + 1|x_j^t)$  for maximizing the error of the domain classifier  $G_{C_1}$  and align the sample towards the ‘unknown’ class.

**Case 3 (A<sub>H</sub>, B<sub>L</sub>):** For this case, the computed similarity measure (Equation (9)) will be low and this will assist the generator in deciding to increase the value of  $P(y = N + 1|x_j^t)$  for maximizing the error of the domain classifier  $G_{C_1}$  and align the sample towards the ‘unknown’ class.

**Case 4 (A<sub>H</sub>, B<sub>H</sub>):** For this case, the computed similarity measure (Equation (9)) will be high and this will assist the generator in deciding to decrease the value of  $P(y = N + 1|x_j^t)$  for maximizing the error of the domain classifier  $G_{C_1}$  and align the sample towards the known classes.

We train the supplementary domain classifier  $G_D$  as follows,

$$\begin{aligned} E_{G_D} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log(G_D(G_F(x_i^s))) \\ -\frac{1}{n_t} \sum_{j=1}^{n_t} \log(1 - G_D(G_F(x_j^t))). \end{aligned} \quad (10)$$

Equations (9) and (10) indicate that the outputs of  $G_D$  are dependent on the output of the supplementary source classifier  $G_{C_2}$  and the output of the domain classifier  $G_{C_1}$ . This verifies that  $G_D$  is trained to evaluate target samples based on the discriminative known classes and unknown class label information, which will assist  $G_D$  to assign meaningful and identifiable weights to target samples belonging to  $C$  and  $\overline{C}_t$ . Thus, we obtain weights to quantify the similarity of target samples to the source domain from  $G_D$  as,

$$W(x_j^t) = G_D(G_F(x_j^t)). \quad (11)$$

During the early phase of training, if either one of the classifiers ( $G_{C_1}, G_{C_2}$ ) produces uncertain similarity measure ( $d_2(x), d_1(x)$ ), it will be supported by the other ones’ decision for assigning weights to the target samples. However, over the training epochs, both the classifiers will converge to their optimal value for the feature extractor  $G_F$ . In such an advanced phase of training, the target samples belonging to the shared label space will surely get close to 1 similarity score from  $G_{C_2}$  as it is trained on source samples only. Similarly, the similarity score for that target sample from  $G_{C_1}$  will also be high as  $G_{C_1}$  learns to yield low ‘unknown’ class probability for target samples belonging to  $C$ . Thus, the combined weight  $W(x_j^t)$  will be high, and the sample will be aligned to the known classes. On the other hand, if a target sample comes from outside the shared label space, then the  $G_{C_2}$  will produce close to 0 similarity score. The  $G_{C_1}$  will produce low score as well for such sample, and eventually, the weight  $W(x_j^t)$  will be low, and the sample will be aligned to the ‘unknown’

class. Section IV-F provides pictorial representation of learned weights.

Considering all the above-discussed derivations, we present our proposed adversarial domain adaptation model with multi-classifiers. We denote the parameters of the supplementary source classifier  $G_{C_2}$  as  $\theta_{G_{C_2}}$ . The overall objectives of our proposed method are:

$$\begin{aligned}\theta_{G_{C_1}} &= \operatorname{argmin}_{\theta_{G_{C_1}}} E_{G_{C_1}} + E'_{G_{C_1 \text{ adv}}} \\ \theta_{G_F} &= \operatorname{argmin}_{\theta_{G_F}} E_{G_{C_1}} - E'_{G_{C_1 \text{ adv}}} \\ \theta_{G_{C_2}} &= \operatorname{argmin}_{\theta_{G_{C_2}}} E_{G_{C_2}} + E_D.\end{aligned}\quad (12)$$

During back-propagation, we use a gradient reversal layer [6] to calculate the gradient of  $E'_{G_{C_1 \text{ adv}}}$  efficiently. Unlike prior work [18], our proposed model does not need any prior training on the source dataset. We optimize all the objectives simultaneously in an end-to-end fashion.

3) *Constraining positive transfer*: In this section, we discuss how the proposed method limits the tendency of negative transfer in the OSBP model based on the case discussed in Section III-B, which explains negative transfers in the OSBP model. In contrary to OSBP model, for example, when the domain classifier assigns  $P(y = N+1|x_j^t) = 0.4$  for a target sample, the  $G_D$  in our proposed method will generate a weight  $W(x_j^t)$  for the sample after evaluating its similarity to the source domain based on Equations (5 - 11). In short, we compute the value of  $d_2(x_j^t)$  (Equation (6)) and  $d_1(x_j^t)$  (Equation (6)). The former one is 0.6 in this case (we consider this value as a high as it is closer to 1 than 0), and the latter one can be either high and yield high weight  $W(x_j^t)$  or low leading to low weight  $W(x_j^t)$  based on Equation (11). If the weight  $W(x_j^t)$  is high, the proposed model will assist the generator in aligning the sample to known classes by decreasing  $P(y = N+1|x_j^t)$ . Otherwise, if the weight  $W(x_j^t)$  is low, the sample will be aligned to ‘unknown’ class by maximizing the value of  $P(y = N+1|x_j^t)$ . Thus, our DA model does not participate in negative transfer by aligning unknown samples to known classes.

4) *Testing Phase*: During the training phase, we fulfill our goal to transform the domain classifier  $G_{C_1}$  from source domain classifier to target domain classifier, including the category ‘unknown’ by utilizing  $G_{C_1}$  and  $G_D$  classifiers. In the testing phase, we omit the supplementary classifiers and utilize only the trained feature generator  $G_F$ , and  $G_{C_1}$  to classify test images correctly, as shown in Fig. 2c.

#### IV. EXPERIMENTAL STUDIES

In this section, we describe the datasets, our evaluation details, and the results. We conduct experiments to evaluate our proposed method with contemporary DA methods on four standard datasets.

##### A. Datasets

**Office-31** [31] has 31 categories in three visually distinct domains, namely: amazon (**A**), DSLR (**D**) and webcam (**W**).

This dataset comprises a collection of samples from **amazon.com**, captured samples from DSLR and web camera for DA. We have chosen the first 10 classes as  $C$  and the last 10 classes as ‘unknown’ samples in the target domain  $\bar{C}_t$  for accomplishing six open set DA tasks:  $A \rightarrow W$ ,  $D \rightarrow W$ ,  $W \rightarrow D$ ,  $A \rightarrow D$ ,  $D \rightarrow A$  and  $W \rightarrow A$ . **VisDA2017** [56] poses a special DA setting by focusing on a simulation (rendered 3D images) to real-world DA setting. Game engines generate the samples of source domain while the target domain samples are actual images. This dataset comprises 12 categories. Inline with [18], we have chosen six classes (bicycle, bus, car, motorcycle, train, and truck) as the shared classes  $C$  and the remaining six classes as ‘unknown’ classes in the target domain.

**Office-Home** [57] consists of 65 classes in four different domains: Artistic images (Ar), Clip-Art images (Cl), Product images (Pr), and Real-World images (Rw). The first 10 classes in alphabetical order are used as the shared classes  $C$ . Leaving the next five classes private to the source domain, the rest classes are considered as ‘unknown’ or private to the target domain. For this dataset, we have designed 12 open set DA tasks:  $Ar \rightarrow Cl$ ,  $Ar \rightarrow Pr$ ,  $Ar \rightarrow Rw$ ,  $Cl \rightarrow Ar$ ,  $Cl \rightarrow Pr$ ,  $Cl \rightarrow Rw$ ,  $Pr \rightarrow Ar$ ,  $Pr \rightarrow Cl$ ,  $Pr \rightarrow Rw$ ,  $Rw \rightarrow Ar$ ,  $Rw \rightarrow Cl$  and  $Rw \rightarrow Pr$ . **ImageNet-Caltech** is a combination of ImageNet-1K [58] consisting 1000 categories and Caltech-256 with 256 categories. In line with previous works [20], [21], we have used the common 84 classes as the known or shared classes  $C$  and have used the remaining classes as the ‘unknown’ class in the target domain. We have performed two open set DA tasks  $I \rightarrow C$  and  $C \rightarrow I$  for this dataset.

##### B. Evaluation Details

**Evaluation Protocols.** In this paper, we have followed the evaluation protocol of the Visual Domain Adaptation (VisDA 2018) Open-Set Classification Challenge. This protocol assumes all the target domain private classes  $\bar{C}_t$  as a unified ‘unknown’ class and the average per-class accuracy for all the  $|C|+1$  classes is the final result. Also, being inspired by [31], we present the normalized classification accuracy measured on all the known classes and the ‘unknown’ classes ( $|C|+1$ ) as OS, and the normalized classification accuracy only on the shared classes ( $|C|$ ) as OS\*.

**Implementation Details.** We have used ImageNet pre-trained ResNet-50 and ResNet-152 [2] with new fully-connected and batch normalization layers as the feature generator. We have used SGD with a learning rate of 0.001 for pre-trained layers, 10 times higher than that for new layers, and momentum of 0.9. Note, while the original papers show results on a variety of backbone networks such as VGG, AlexNet, and ResNet-50, for the sake of fairness and consistency we tested them all using ResNet-50. Also, we have executed up to 1000 epochs for training, but if any of the contemporary methods converged earlier than that, we stopped the training. The performance difference between our implementation of the contemporary DA methods and the original papers is mainly due to different backbone networks and the number of iterations. Note that the results cannot be directly compared against other publicly reported results due to different train-test data split and versions of PyTorch.

Table I: Classification accuracy (%) of proposed and contemporary domain adaptation methods on Office-Home dataset tasks.

Approach	Accuracy (%)																									
	Ar→Cl OS OS*		Ar→Pr OS OS*		Ar→Rw OS OS*		Cl→Ar OS OS*		Cl→Pr OS OS*		Cl→Rw OS OS*		Pr→Ar OS OS*		Pr→Cl OS OS*		Pr→Rw OS OS*		Rw→Ar OS OS*		Rw→Cl OS OS*		Rw→Pr OS OS*		Avg	
ResNet [2] (2016)	54.4	54.8	69.5	70.1	78.7	78.1	62.0	61.3	60.8	62.3	71.6	72.5	64.2	64.4	58.9	58.6	75.5	76.1	70.3	69.2	52.5	51.5	74.5	75.3	66.1	66.2
DANN [29] (2016)	-	44.8	-	68.5	-	79.5	-	65.5	-	57.9	-	67.4	-	56.9	-	40.2	-	77.5	-	68.5	-	45.2	-	77.6	-	62.4
RTN [16] (2016)	-	50.9	-	75.6	-	82.9	-	66.5	-	73.4	-	85.7	-	65.6	-	47.9	-	84.5	-	78.1	-	56.9	-	77.6	-	70.4
IWAN [22] (2018)	53.1	52.1	79.4	78.5	86.1	86.4	70.2	69.7	70.9	71.3	86.8	85.1	74.9	74.5	55.6	55.7	85.1	84.2	77.9	78.7	60.8	59.4	77.2	76.8	73.1	72.7
ETN [20] (2019)	59.3	59.0	77.1	76.8	85.6	85.7	63.1	62.9	65.6	65.1	75.3	75.6	68.3	67.8	55.4	55.6	86.4	85.9	78.7	77.5	62.3	61.5	84.4	84.2	71.8	71.4
UAN [24] (2019)	63.0	62.5	82.8	82.4	86.8	85.9	76.8	76.9	78.7	79.1	84.4	84.8	78.2	77.4	58.6	57.8	86.8	85.9	82.5	82.5	63.2	63.0	79.1	78.1	76.8	76.6
BP [6] (2015)	53.6	51.0	69.1	65.9	75.9	74.1	59.5	57.3	65.2	62.5	73.2	72.0	47.2	45.0	43.9	40.2	78.7	76.4	70.6	65.3	45.6	42.1	77.5	74.2	63.3	60.5
ATI [19] (2017)	53.8	51.3	80.4	77.9	86.1	85.0	71.2	67.8	72.3	70.5	85.1	83.2	74.3	72.5	57.9	55.1	85.6	84.7	76.1	75.2	60.2	58.7	78.3	77.0	73.4	71.5
OSBP [18] (2018)	48.5	48.6	70.9	70.6	75.2	74.2	59.5	58.2	61.6	59.9	75.1	74.5	61.9	62.2	43.5	43.2	79.9	80.4	70.1	70.2	53.9	54.1	75.7	75.4	64.6	64.3
STA [25] (2019)	57.8	58.1	71.3	70.1	84.9	85.5	61.4	61.9	68.1	67.9	75.2	75.8	64.3	63.2	51.8	52.2	80.2	79.1	73.9	74.2	53.6	54.5	80.5	81.4	68.6	68.7
Our(ResNet-50)	64.8	64.9	84.6	84.9	<b>88.1</b>	<b>88.2</b>	79.6	79.9	81.6	82.9	85.6	86.8	77.9	78.5	61.8	63.1	85.9	87.5	85.4	85.6	67.5	65.2	80.9	80.1	78.6	79.0
Our(ResNet-152)	<b>66.1</b>	<b>66.9</b>	<b>85.9</b>	<b>86.7</b>	87.6	87.9	<b>80.5</b>	<b>80.8</b>	<b>83.5</b>	<b>85.6</b>	<b>86.9</b>	<b>87.9</b>	<b>79.9</b>	<b>81.1</b>	<b>63.1</b>	<b>64.8</b>	<b>87.9</b>	<b>88.7</b>	<b>86.9</b>	<b>88.2</b>	<b>68.8</b>	<b>69.7</b>	<b>82.1</b>	<b>82.8</b>	<b>80.0</b>	<b>80.9</b>

Table II: Classification accuracy (%) of proposed and other domain adaptation methods on Office-31 tasks.

Approach	Accuracy (%)													
	A→W OS OS*		D→W OS OS*		W→D OS OS*		A→D OS OS*		D→A OS OS*		W→A OS OS*		Avg	
ResNet [2] (2016)	83.2	83.8	93.8	94.7	95.8	94.6	84.6	84.7	72.3	71.9	75.5	75.3	84.2	84.2
DANN [29] (2016)	-	80.2	-	79.9	-	87.5	-	80.4	-	74.9	-	80.9	-	80.6
RTN [16] (2016)	-	88.1	-	89.8	-	84.8	-	73.1	-	85.1	-	84.7	-	84.2
IWAN [22] (2018)	86.5	84.9	89.9	87.1	91.1	90.3	82.3	79.6	82.2	80.6	85.7	83.1	86.3	84.3
ETN [20] (2019)	85.7	84.4	93.6	92.9	96.9	96.0	84.9	85.1	84.9	85.6	85.2	85.0	88.5	88.1
UAN [24] (2019)	85.6	84.3	94.7	93.9	97.9	97.5	86.5	84.9	85.5	85.6	85.1	84.6	89.2	88.4
BP [6] (2015)	75.9	74.1	89.7	87.2	94.4	93.2	78.4	76.8	56.8	55.3	62.9	62.7	76.3	74.8
ATI [19] (2017)	78.4	74.9	92.6	90.6	97.1	95.6	78.9	77.5	71.6	70.1	76.8	74.2	82.5	80.4
OSBP [18] (2018)	67.4	66.9	83.7	83.5	95.1	94.8	82.6	82.0	76.6	76.5	79.5	78.9	80.8	80.4
STA [25] (2019)	88.6	90.1	97.1	95.2	97.3	97.5	<b>91.9</b>	<b>93.1</b>	88.3	88.6	84.1	84.2	91.2	91.4
Our(ResNet-50)	88.3	88.4	97.3	97.8	98.1	98.4	87.8	88.6	89.9	89.6	84.9	85.8	91.1	91.5
Our(ResNet-152)	<b>90.2</b>	<b>90.8</b>	<b>97.9</b>	<b>98.8</b>	<b>98.6</b>	<b>98.8</b>	89.5	89.7	<b>91.0</b>	<b>91.7</b>	<b>86.8</b>	<b>87.6</b>	<b>92.4</b>	<b>92.9</b>

Table III: Classification accuracy (%) of proposed and other domain adaptation methods on ImageNet-Caltech tasks.

Approach	Accuracy (%)					
	I→C OS OS*		C→I OS OS*		Avg OS OS*	
ResNet [2]	75.7	75.1	67.1	67.8	71.4	71.5
DANN [29] (2016)	-	71.1	-	65.9	-	68.5
RTN [16] (2016)	-	71.9	-	66.2	-	69.1
IWAN [22] (2018)	74.1	72.9	68.7	65.3	71.4	69.1
ETN [20] (2019)	74.9	74.8	69.8	69.9	72.4	72.4
UAN [24] (2019)	75.3	76.3	70.2	70.8	72.7	73.6
BP [6] (2015)	68.9	67.3	61.2	59.0	65.0	63.2
ATI [19] (2017)	71.6	65.9	67.4	65.1	69.5	65.5
OSBP [18] (2018)	63.1	63.4	54.8	53.6	58.9	58.5
STA [25] (2019)	75.3	74.2	68.1	68.3	71.7	71.3
Our(ResNet-50)	77.4	77.8	69.8	70.1	73.6	74.0
Our(ResNet-152)	<b>78.9</b>	<b>79.7</b>	<b>71.9</b>	<b>71.7</b>	<b>75.4</b>	<b>75.7</b>

**Compared Domain Adaptation Methods.** For the sake of thoroughness, we have compared the performance of the proposed method with: **1) Classifier without DA:** ResNet [2] (Note that Negative transfer is calculated against this non-DA classifier); **2) Closed-set DA methods:** Domain-Adversarial Neural Networks (DANN) [29] and Residual Transfer Networks (RTN) [16]; **3) Partial DA methods:** Importance Weighted Adversarial Nets (IWAN) [22] and Example Transfer Network (ETN) [20]; **4) Open set DA methods:** Unsupervised domain adaptation by back-propagation (BP) [6] with unknown source samples, Assign-and-Transform-Iteratively (ATI) [19], Open Set domain adaptation by Back-Propagation (OSBP) [18] and Separate to Adapt (STA) [25]; **5) Universal DA method:** Universal Adaptation Network (UAN) [24].

### C. Classification results

The classification results on the 12 tasks of Office-Home, 6 tasks of Office-31, the task of VisDA and 2 large-scale tasks of ImageNet-Caltech are shown in Tables I, II, III and IV respectively ('-' indicates that results could not be regenerated because of closed set DA setting and negative transfer for DA classifiers against the non-DA classifier ResNet [2] (shown in gray background) are indicated by showing the classification accuracy in italic). Our proposed method outperforms all the compared methods in terms of the average per-class accuracy except for Office-31 dataset, where STA [25] leads by 0.1%. However, we have better OS\* than STA for Office-31 dataset. We observe that all contemporary partial, universal, and open set DA methods lag behind ResNet classifier [2] on some tasks because of negative transfer during adaptation. This negative transfer is the effect of the difference in source and target domain label space introduced by unknown classes. In addition, the accuracy of the OS\* is lower than that of OS for the majority of the tasks, which means a large number of unknown images are misclassified. Our proposed method yields better results for OS\* than OS which indicates a better separation of known and unknown target samples.

DA methods such as BP (with unknown classes in the source domain) [6] and ATI [19] are trained to align target domain towards source domain employing distribution matching methods. During this process, the unknown source or target samples disturb the known class feature alignment leading to such performance degradation. The performance lag in OSBP [18] method compared to ResNet backbone for some tasks supports our claim. That is, the OSBP incurs negative transfer because it tends to align some unknown target samples to known classes

Table IV: Classification accuracy (%) of proposed and other domain adaptation methods on VisDA2017 tasks.

Approach	Accuracy (%)								
	bicycle	bus	car	motorcycle	train	truck	unknown	OS	OS*
ResNet [2] (2016)	40.2	55.4	63.5	70.8	74.1	35.2	45.6	54.9	56.5
DANN [29] (2016)	32.4	51.6	65.1	71.3	85.1	23.1	-	-	52.1
RTN [16] (2016)	31.6	63.6	54.2	76.9	<b>87.3</b>	21.5	-	-	51.1
IWAN [22] (2018)	30.6	69.8	58.3	76.8	65.5	30.8	69.7	57.3	55.3
ETN [20] (2019)	31.6	66.8	61.7	77.8	70.8	30.8	70.7	58.6	56.6
UAN [24] (2019)	42.6	67.8	65.7	76.9	69.8	31.8	70.7	60.9	59.1
BP [6] (2015)	31.8	66.5	50.5	70.1	86.9	21.8	38.5	52.3	54.6
ATI [19] (2017)	33.6	51.6	64.2	78.1	85.3	22.5	42.5	54.8	52.6
OSBP [18] (2018)	35.6	59.8	48.3	76.8	55.5	29.8	81.7	55.4	50.9
STA [25] (2019)	50.1	69.1	59.7	<b>85.7</b>	84.7	25.1	<b>82.4</b>	65.3	62.4
Our(ResNet-50)	50.6	74.8	66.7	80.6	75.9	38.8	73.9	65.9	64.6
Our(ResNet-152)	<b>52.1</b>	<b>77.7</b>	<b>67.7</b>	81.4	80.8	<b>39.8</b>	75.5	<b>67.8</b>	<b>66.6</b>

by enforcing empirical boundary threshold, and not exploring underlying domain information of target samples. However, because of adapting sample-level transferability, IWAN [22], ETN [20], UAN [24], and STA [25] have lower negative transfer rate compared to other existing methods.

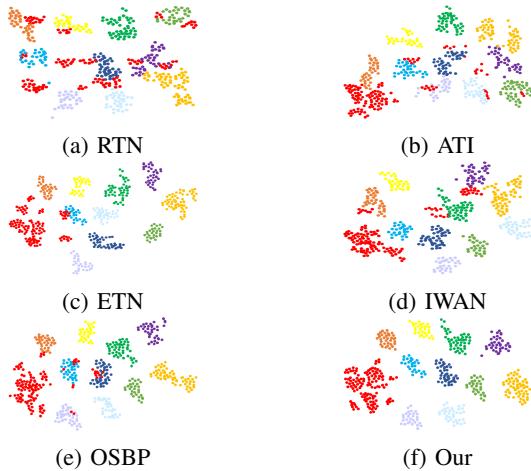


Figure 3: Learned features of our proposed method for  $A \rightarrow W$  task from **Office-31** dataset show a better separation of unknown samples (red dots) from known classes than contemporary DA methods. We select 10 shared classes, 10 source domain private classes, and 10 target domain private classes for the task. Visualization prepared by the t-SNE algorithm [59] with the Perplexity parameter set to 50.

In Fig. 3, we plot the t-SNE [59] embeddings of the features learned by RTN [16], ATI [19], IWAN [22], ETN [20], OSBP [18] and proposed method on  $A \rightarrow W$  task with 10 shared classes, 10 source domain private classes and 10 target domain private classes as per respective DA settings. The proposed method demonstrates significantly comprehensible and well-segregated clusters for all known and unknown classes than other DA methods. This distinct separation of unknown target samples from the known ones is because of the supplementary source classifier, which is trained only on the source samples to learn discriminative features for known classes. Unlike OSBP [18], we do not need any prior training on the source domain

to learn discriminative known class features to support better classification. On the other hand, RTN [16] and ATI [19] methods which utilize distribution matching techniques such as MMD only tries to align target samples with the source ones and do not separate well among known and unknown classes. Though ETN [20] shows better clusters than IWAN [22] for initiating less negative transfers, it certainly lags behind our method.

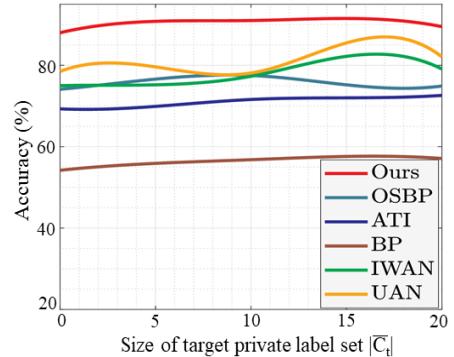


Figure 4: The proposed method consistently performs better than contemporary domain adaptation methods for all cases of  $|C_t|$ . The performance of our proposed method increases with an increase of the number of unknown classes in the target domain  $|C_t|$ .

#### D. Analysis on Various Open Set DA Settings

**Varying Size of  $\bar{C}_t$ .** We compare the performance of our proposed method with other methods by varying the number of unknown samples in the target domain for  $D \rightarrow A$  task. Fig. 4 shows that our proposed method maintains moderate increment of performance with no significant drop in-between transitions of varying  $|C_t|$  and outperforms all compared DA methods for all cases. This indicates a larger number of unknown classes in the target domain  $|C_t|$  compared to the shared classes assists both the source  $G_{C_1}$  and supplementary source  $G_{\bar{C}_t}$  classifiers, by initiating less distraction and can lead to solving more realistic tasks where unknown source samples are unavailable. Note that our proposed method does not take advantage of

Table V: Accuracy (%) of proposed method on VisDA2017, Office-31, and ImageNet-Caltech tasks for ablation study.

Approach	Accuracy (%)										
	VisDA	A → W	D → W	W → D	A → D	D → A	W → A	Avg	I → C	C → I	Avg
w/o $d_2$	63.3	86.1	93.1	96.2	84.2	86.5	82.1	88.0	75.3	66.4	70.9
w/o $d_1$	61.1	84.8	93.8	94.7	85.1	84.7	82.3	87.6	74.6	65.0	69.8
Our(ResNet-50)	65.9	88.3	97.3	98.1	87.8	89.9	84.9	91.1	77.4	69.8	73.6

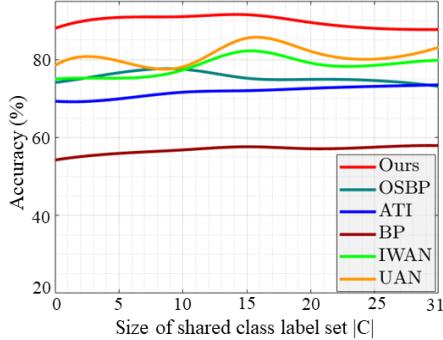


Figure 5: Our proposed method outperforms contemporary domain adaptation methods with different domain adaptation settings for different sizes of the shared label set. We observe that when  $|C|$  reaches beyond 20, the performance of our method decreases slightly. This indicates our proposed method is more suitable for tasks that have more ‘unknown’ classes in the target domain.

any prior knowledge about the label sets like OSBP [18] and IWAN [22].

**Varying Size of  $C$ .** We further explore the performance of the proposed method by varying the number of shared classes for the same task  $D \rightarrow A$  and compare it with other methods (Fig. 5). The proposed method maintains high accuracy with a slight drop when  $|C| > 20$ . The evaluation of the task shown here is a sub-task of the Office-31 dataset. The office-31 dataset has 31 categories when the shared label set  $C$  is beyond 20 the target private label set  $|\bar{C}_t|$  decreases to less than 10. The proposed method is designed to handle well the tasks which have a large number of unknown classes in the target private label space. Therefore, the increment of shared label space, which causes decrement of target private label space in a large proportion, harms the model. However, the proposed method substantially outperforms other compared methods for all cases of  $|C|$ .

### E. Ablation Study

We execute an ablation study for evaluating two variants of the proposed domain adaptation method with the multi-classifier based weighting module to investigate deeper into its effectiveness. **w/o  $d_2$**  is the variant of proposed method without integrating the outcome of  $d_2(x)$  (domain information) in the procedure of weighting target samples, i.e. in Equations (9) and (10). **w/o  $d_1$**  is the variant without integrating the known-source label information into the weighting mechanism by omitting  $G_{C_2}$  and deploying  $G_D$  to depend only on the value of  $d_2(x)$  for source and target samples. To execute this

variant, we need to omit Equations (5) and (7), omit  $d_1(x)$  in Equations (9) and (10).

Table V presents the results for the variants of the proposed method, as mentioned above. The performance of both **w/o  $d_2$**  and **w/o  $d_1$**  lag behind the proposed method. This is because both the deciding factors  $d_1(x_i^t)$  and  $d_2(x_i^t)$  are required for defining meaningful weights  $W(x_j^t)$  to the target samples. We also observe that the variant without **w/o  $d_2$**  achieves better average per-class accuracy than the other one for all three tasks, which indicates integrating the supplementary source classifier  $G_{C_2}$  in the weighting module for exploiting label information is more effective.

### F. Weight Visualization

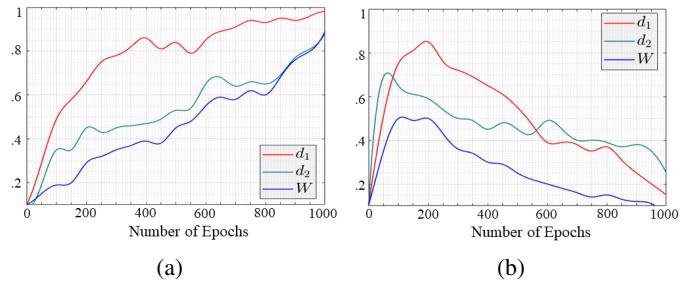


Figure 6: Pictorial representation of learned weights of (a) known target samples and (b) unknown target samples for the task  $D \rightarrow A$  (Office-31). Here,  $W$  represents the weights assigned to target samples during training by  $G_D$  after assessing the two similarity measures  $d_1$  and  $d_2$ .

To analyse the trend of generated weights, we plot the learned similarity measures ( $d_1$  and  $d_2$ ) and final weights ( $W$ ) of known and unknown target samples against the training epochs in Figure 6. Figures 6a and 6b show that our proposed method assigns sufficiently high and low weights ( $W$ ) to known and unknown target samples, respectively. It is evident that such weights will assist the adversarial module in enhancing positive transfer by better separation of known target samples from unknowns. During the early stage of learning, both  $d_1$  and  $d_2$  increases for known and unknown samples which means the classifiers give uncertain decisions. However, after some initial epochs, for known samples,  $d_1$  seems to increase at a larger pace and learns to yield a much higher value than  $d_2$ . This is because the classifier  $G_{C_2}$  is trained only on source samples. On the other hand, for unknown samples, both the similarity measures start to decrease. When both  $G_{C_2}$  and  $G_{C_1}$  converge to their optimal value for  $G_F$ ,  $d_1$ ,  $d_2$  and  $W$  for known target samples reaches near 0.98, 0.88 and 0.86 respectively. Which means the target samples get very high weights as the training proceeds and dictate  $G_F$ .

to decrease the ‘unknown’ class probability for aligning them to known classes. For unknown target samples,  $d_1$ ,  $d_2$  and  $W$  decreases to 0.15, 0.25 and 0.04 respectively. This indicates the target samples get very low weights and assists  $G_F$  to increase the ‘unknown’ class probability and align them to unknown classes. It is worth mentioning, the final value of  $d_1$ ,  $d_2$  and hence  $W$  may differ a little based on the dataset or task.

## V. CONCLUSION

In this paper, we propose a multi-classifier based domain adversarial network for an open set domain adaptation setting, where the target domain has a larger number of classes than the source domain. In our proposed method, the multi-classifier structure poses a weighting module that explores discriminative label and domain information, and assign distinguishable scores to the known and unknown target samples for enhancing positive transfer and eventually, assisting the feature generator and the domain classifier to separate known and unknown target samples. Another noteworthy attribute of our proposed method is the ability to discover the boundary between shared label space and target private label space automatically. A thorough experimental evaluation has demonstrated that the proposed method consistently outperforms the existing DA methods. We have further shown through the ablation study that the two deciding factors for generating weights for target samples are crucial for maintaining the integrity of the model and initiating positive sample transfer.

## REFERENCES

- [1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.
- [4] T. Shermin, S. W. Teng, M. Murshed, G. Lu, F. Sohel, and M. Paul, “Enhanced transfer learning with imagenet trained classification layer,” in *Proceedings of the Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, 2019.
- [5] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3722–3731.
- [6] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International Conference on Machine Learning (ICML)*, 2015, pp. 1180–1189.
- [7] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, “Heterogeneous domain adaptation through progressive alignment,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1381–1391, 2018.
- [8] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, “Locality preserving joint transfer for domain adaptation,” *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6103–6115, 2019.
- [9] X. Ma, T. Zhang, and C. Xu, “Deep multi-modality adversarial networks for unsupervised domain adaptation,” *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2419–2431, 2019.
- [10] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7167–7176.
- [11] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 700–708.
- [12] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” *International Conference on Learning Representations (ICLR)*, 2016.
- [13] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3723–3732.
- [14] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, “Transfer independently together: A generalized framework for domain adaptation,” *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2144–2155, 2018.
- [15] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2014.
- [16] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Unsupervised domain adaptation with residual transfer networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 136–144.
- [17] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers, “Associative domain adaptation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2765–2773.
- [18] K. Saito, S. Yamamoto, and Y. U. T. Harada, “Open set domain adaptation by backpropagation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 153–168.
- [19] P. P. Bustos and J. Gall, “Open set domain adaptation,” in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2017, pp. 754–763.
- [20] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, “Learning to transfer examples for partial domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2985–2994.
- [21] Z. Cao, M. Long, J. Wang, and M. I. Jordan, “Partial transfer learning with selective adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2724–2732.
- [22] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, “Importance weighted adversarial nets for partial domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8156–8164.
- [23] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [24] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Universal domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [25] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, “Separate to adapt: Open set domain adaptation via progressive separation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2927–2936.
- [26] Z. Cao, L. Ma, M. Long, and J. Wang, “Partial adversarial domain adaptation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 135–150.
- [27] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *Advances in Neural Information Processing Systems*, 2018, pp. 1640–1650.
- [28] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” *arXiv preprint arXiv:1502.02791*, 2015.
- [29] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research (JMLR)*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [30] J. Hoffman, E. Tzeng, T. Darrell, and K. Saenko, “Simultaneous deep transfer across domains and tasks,” in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 173–187.
- [31] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 213–226.
- [32] X. Wang and J. Schneider, “Flexible transfer learning under support and model shift,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1898–1906.
- [33] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.
- [34] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2066–2073.

- [35] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *International Conference on Machine Learning*, 2013, pp. 819–827.
- [36] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.
- [37] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 2208–2217.
- [38] W. Zellinger, T. Grubinger, E. Lughofner, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," *arXiv preprint arXiv:1702.08811*, 2017.
- [39] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 443–450.
- [40] 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*, 2014, pp. 2672–2680.
- [41] R. Volpi, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5495–5504.
- [42] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.
- [43] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4500–4509.
- [44] Y.-C. Liu, Y.-Y. Yeh, T.-C. Fu, S.-D. Wang, W.-C. Chiu, and Y.-C. Frank Wang, "Detach and adapt: Learning cross-domain disentangled deep representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8867–8876.
- [45] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, "Auggan: Cross domain adaptation with gan-based data augmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 718–731.
- [46] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [47] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: symmetric bi-directional adaptive gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8099–8108.
- [48] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *arXiv preprint arXiv:1711.03213*, 2017.
- [49] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and Z. Huang, "Cycle-consistent conditional adversarial transfer networks," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 747–755.
- [50] F. M. Cariucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Autodial: Automatic domain alignment layers," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 5077–5085.
- [51] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.
- [52] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European Conference on Computer Vision*. Springer, 2014, pp. 393–409.
- [53] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1563–1572.
- [54] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," *arXiv preprint arXiv:1707.07418*, 2017.
- [55] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 2642–2651.
- [56] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko, "Visda: A synthetic-to-real benchmark for visual domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2021–2026.
- [57] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5018–5027.
- [58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [59] L. Maaten and G. Hinton, "Visualizing data using t-sne," *The Journal of Machine Learning Research (JMLR)*, vol. 9, no. Nov, pp. 2579–2605, 2008.



**Tasnia Shermin** has received a bachelor's degree in Computer Science and Engineering (CSE) from the Military Institute of Science and Technology, Bangladesh, in 2018. She is currently pursuing the Doctor of Philosophy (Ph.D.) degree at the School of Science, Engineering and Information Technology, Federation University, Australia. Her research interests include artificial intelligence in computer vision, deep learning and transfer learning.



**Guojun Lu** is a Professor in the School of Science, Engineering and Information Technology, Federation University Australia. He has many years' research experience in artificial intelligence, multimedia signal processing and retrieval, and has successfully supervised over 20 PhD students. He has held positions at Loughborough University, National University of Singapore, Deakin University and Monash University, after he obtained his PhD from Loughborough University and BEng from Nanjing Institute of Technology (now South East University, China). He has published over 230 refereed journal and conference papers and wrote two books.



**Shyh Wei Teng** is an Associate Professor and Deputy Dean at School of Science, Engineering and IT, Federation University Australia. His research interests include Image/video processing; Machine learning; and Multimedia analytics. He has so far published over 70 refereed research papers. He received various competitive research funding at state, federal and international levels.



**Manzur Murshed** (M'96–SM'12) received a BScEngg (Hons) degree in computer science and engineering from Bangladesh University of Engineering and Technology in 1994 and a PhD degree in computer science from the Australian National University in 1999. Currently, he is a Professor and the Associate Dean (Research) at the School of Science, Engineering and Information Technology, Federation University Australia. His research interests include video technology, machine learning, wireless communications, Cloud computing, and security privacy. He has published 230+ refereed research papers, received \$5m research grants, and supervised 25 PhDs. He is serving IEEE TMM and served IEEE TCSVT as an Associate Editor.



on Multimedia.

**Ferdous Sohel** (M'08–SM'13) received his PhD degree from Monash University, Australia, in 2009. He is currently an Associate Professor in Information Technology at Murdoch University, Australia. Prior to his joining Murdoch University, he was a Research Assistant Professor at the University of Western Australia from 2008 to 2015. His research interests include computer vision, image processing, machine learning, digital agritech, health analytics, cyber forensics, and video coding. He has been serving as an Associate Editor of IEEE Transactions