Open Set Domain Adaptation with Multi-Classifier Adversarial Network

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

Abstract—**Domain adaptation aims to transfer knowledge from a domain with adequate labeled samples to a domain with scarce labeled samples. The majority of existing domain adaptation methods rely on the assumption of having identical label spaces across the source and target domains, which limits their application in real-world scenarios. To get rid of such an assumption, prior research has introduced various open set domain adaptation settings in the literature. This paper focuses on the type of open set domain adaptation setting where the target domain has both private ('unknown classes') label space beside 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 with an empirical fixed threshold which lacks at handling false-negative transfers. We propose a multi-classifier based weighting scheme for the adversarial domain adaptation model to address this issue and improve performance. Our proposed method assigns distinguishable weights to target samples belonging to the known and unknown classes to limit false-negative transfers, and simultaneously reduce the domain gap between shared classes of the source and target domains. A thorough evaluation shows that our proposed method outperforms existing domain adaptation methods for a number of domain adaptation datasets.**

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

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, the performance of this approach may suffer. Also, synthesizing data by using different variants of sensors may create domain divergence. Existing domain adaptation (DA) methods aim to decrease the above-mentioned domain divergences either by using distribution matching methods [5], [6], [7] or by transforming samples from one domain to another through generative models [8], [5], [9], [10], [11].

Generally, it is assumed that label sets across the source and target domains are identical (closed set domain adaptation [12], [13], [14]), as shown in Fig. 1a, which is not always the case in real-world scenarios. Open set domain adaptation [15], [16] and partial domain adaptation [17], [18], [19] 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 [17], [18], [19] setting assumes that the source domain label space is a superspace of the target domain label space. Open set domain adaptation setting proposed by Busto et al. [16] leverages the presence of images from unknown classes in the source domain to align towards the images of unknown classes in the target domain (Fig. 1c). However, it is not a cost-effective way to perform 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, Busto et al. [16] 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.

On the other hand, open set domain adaptation by back-propagation (OSBP) [15] setting takes another step towards practical domain adaptation scenario and removes unknown classes from the source domain such that the source label space is a subset of the target label space (Fig. 1d). By this means, 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 sample as unknown when it does not belong to the known classes. Thus, this setting trains the model with broader unknown class boundary than the previous setting [16] to better detect unknown samples during testing. This paper focuses on improving the performance of domain adaptation model for 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.

In the OSBP domain adaptation setting, it is required to recognize the target samples not belonging to the shared label space as 'unknown' and align the target samples from known classes towards the known classes of the source domain. Prevalent distribution matching domain adaptation methods [5], [6], [7] cannot be applied to OSBP domain adaptation setup as the absence of unknown samples in the source domain

(a) Closed set domain adaptation

(b) Partial domain adaptation

(c) Open set domain adaptation [16]

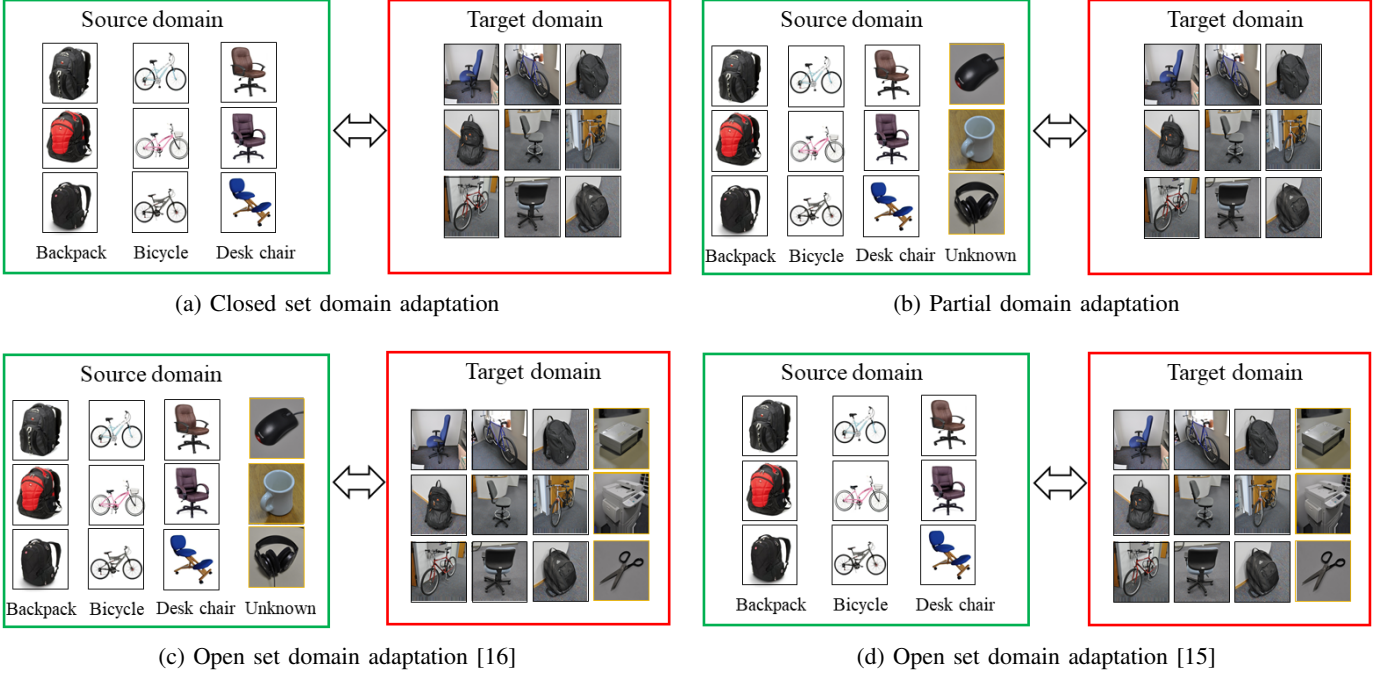(d) Open set domain adaptation [15]

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. This setting does not require images of unknown classes in either domain. (b) Partial domain adaptation setting assumes that the source domain label space (classes) is the superspace of the target domain label space. This domain adaptation setting includes images of unknown classes in the source domain only. (c) Open set domain adaptation setting proposed by Busto et al. [16] 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. [15] requires images of unknown classes only in the target domain besides the classes of interest. Partial (b) and open set ((c) and (d)) domain adaptation settings represent valuable progress towards practical domain adaptation scenarios. We focus on (d) domain adaptation setting in this work.

does not allow the unknown samples of the target domain to be aligned.

Saito et al. [15] have proposed a generative model to address the OSBP domain adaptation setting where they enable the source classifier to draw a boundary between the known and unknown target samples, and the generator to generate target samples such that the samples lie far from the boundary of known samples. They set an empirical fixed threshold for training the source classifier adversarially to differentiate known target samples from unknown ones. However, we argue that this concept of relying only on the decision of the source classifier without considering underlying domain discriminative characteristics may initiate negative transfer of target samples.

To this end, we propose a multi-classifier based adversarial learning domain adaptation model. By utilizing multiple classifiers or discriminators, our proposed model evaluates the transferability of the target samples based on their similarities to the source domain (domain information), resemblance to each source domain classes (discriminative known-source label information), and the probability of belonging to unknown or target private classes (target samples which do not belong to the known classes of the source domain); this constitute our multi-classifier based weighting scheme. The mentioned deciding factors ensure that the proposed weighting scheme

assigns identifiable scores to the target samples coming from the shared label space and target private label space and helps the model to align them towards known classes of the source domain or reject as 'unknown' class. In particular, the proposed model generates weights for target samples by exploring underlying domain discriminative factors and improves performance over previous work [15] by reducing negative target sample transfers.

To the best of our knowledge, we are the first in attempting to integrate a multi-classifier based weighting scheme in an adversarial domain adaptation method for the open set domain adaptation setting, which has unknown class images in the target domain only. A few attempts to address partial domain adaptation [19], [17], and universal domain adaptation [20] by generating weights are identified among the literature. Zhang et al. [19] utilize the Sigmoid 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 domain adaptation. For improving prior partial domain adaptation methods prone to negative transfer [18], [18], Example Transfer Network (ETN) [17] 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, UAN [20] integrates domain similarity and prediction uncertainty. All of these methods concentrate on assigning weights to the source samples, whereas we assign weights to the target samples as per the demand of the OSBP domain adaptation setting. It is worth mentioning, assigning weights to the target samples is more challenging than source samples because we do not have knowledge of the labels of target samples and shared label space during training.

The main contributions of this paper are as follows:

- We propose a multi-classifier based adversarial domain adaptation model 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 evaluates several fundamental domain discriminative information to assign identifiable weights to target samples to enhance positive transfer of target samples during training.
- We execute comprehensive experiments and demonstrate that our proposed model reduces the rate of negative transfer and achieves better performance than contemporary domain adaptation works on several datasets.

Section II presents a brief discussion about contemporary domain adaptation methods. Details on our proposed method are described in Section III. Thorough performance comparison and analysis of the proposed method with contemporary domain adaptation approaches on several 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 domain adaptation methods concentrate on reducing the divergence between the source and the target domains. Recent works have shown that because of its setting, closed set domain adaptation 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 domain adaptation methods [21], [6], [13], [12], [9], [22], [23], [14], [24] is originated from prior shallow domain adaptation methods [25], [26], [27], [28], [29], [30].

Closed set domain adaptation methods fall into three main categories. The first category of methods is based on static moment matching, such as Maximum Mean Discrepancy (MMD) [31], [12], [13], [22], Central Moment Discrepancy (CMD) [32], and second-order statistics matching [33]. The second category of methods adapts the adversarial loss concept of GAN [34] and initiates the generation of images that are non-discriminative to the shared label space of source and target domain [24], [7], [9]. Furthermore, domain adversarial methods align pixels and features from both domains and synthesize labeled target images for data augmentation [35], [36], [37], [38], [39], [5]. Recent research has explored Cycle-Consistent Generative Adversarial Network [40] for developing CycleGAN-based [41], [42] domain adaptation methods. The final category of methods leverages Batch Normalization statistics for aligning the feature distributions of source and target domains to a canonical cone [43], [44].

Closed set domain adaptation methods emphasize on addressing the fundamental issues in distribution matching and at the same time offers scope for extending domain adaptation.

B. Partial Domain Adaptation

The vanilla closed set domain adaptation's assumption that the source and target domains share the same label space does not hold in partial domain adaptation. The setting of partial domain adaptation assumes a target domain that has a fewer number of classes than the source domain. Cao et al. [18] use multiple domain discriminators along with class-level and instance-level weighting mechanism to obtain class-wise adversarial distribution matching for solving partial domain adaptation.

Cao et al. [45] refine Selective Adversarial Network (SAN) [18] by using only one adversarial network and integrating the class-level weight to the source classifier. The aforementioned methods on partial domain adaptation present a step towards a more practical domain adaptation setting.

C. Open Set Domain Adaptation

Open set domain adaptation 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 [46] 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. [47] integrates 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. [48] 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, open set domain adaptation proposed by Busto et al. [16] 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 domain adaptation method takes advantage of the target samples from unknown classes to align with source samples from unknown classes. Saito et al. [15] modified the open set domain adaptation setting for addressing more practical adaptation cases. Their modified setting does not require unknown classes in the source domain. Here, open set domain adaptation is addressed by adversarially training a classifier with an extra category named 'unknown'. However, their proposed method utilizes an empirical threshold for adversarial training that may harm the training procedure of the generator in some cases. Our proposed method overcomes

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

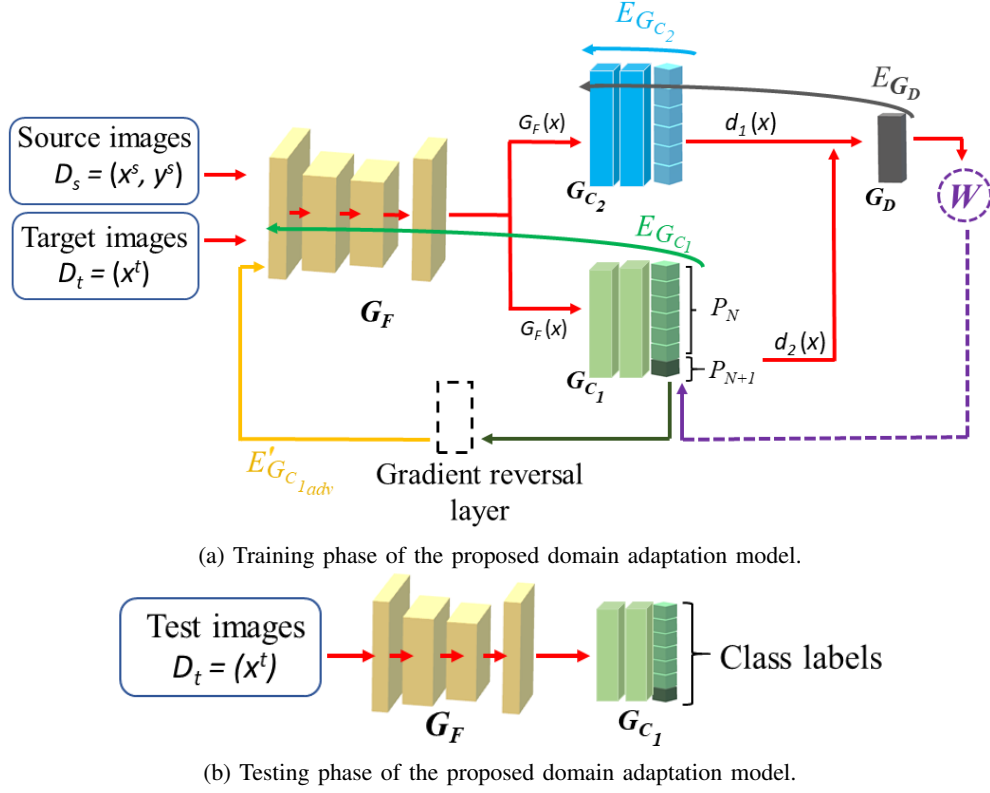(b) Testing phase of the proposed domain adaptation model.

Figure 2: (a) 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 D denotes the generator, domain classifier, 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 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. (b) A pictorial illustration of the proposed model during the testing phase. Red arrows denote forward passes while other arrows represent a backward pass.

such shortcomings and improves classification performance over their method.

D. Universal Domain Adaptation

Recent research introduces a universal domain adaptation setting that imposes no previous knowledge on the source, and target domain label sets [20]. This is another valuable step towards addressing a practical adaptation scenario. Authors propose a Universal Adaptation Network (UAN), which combines domain similarity and prediction uncertainty while generating sample weights for finding shared label sets.

III. PROPOSED METHOD

In this section, we formally describe the open set domain adaptation setting, which is the focus of this work, limitations of OSBP method [15] and present our proposed method.

Problem setup: 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 domain as C_t and C_s respectively. The shared label space is denoted as $C = C_s \cap C_t$. $\overline{C_t} = C_t \setminus C$ 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.

A. Training Phase

1) Domain Adversarial Model: The proposed method is illustrated in Fig. 2a. We aim 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 [23], [24], [15]. The first player of our model is a domain classifier G_{C_1} that is trained to distinguish the features of the source domain from the target domain. The second player is a feature generator G_F that is simultaneously trained to reduce feature distribution divergence in the opposite direction of the domain classifier. The ultimate goal is to train a source domain classifier that is transferred to the target domain classifier with an extra category named 'unknown'. Unlike contemporary adversarial domain adaptation works [17], [19],

[20], we have trained our domain classifier G_{C_1} to act as the source classifier.

The feature generator G_F takes inputs from both source domain D_s and target D_t domain at the same time. In line with OSBP method [15], our 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 forward pass, within G_{C_1}, the features are transformed to a $N+1$-dimensional class probability through softmax function as,

$$\sigma(z) = \frac{\exp(z)}{\sum_{i=1}^{N+1} \exp(z_i)} \qquad (1)$$

where z is the logit vector. Our domain classifier G_{C_1} has to distinguish between known and unknown target samples as well. As the target domain is unlabeled during training, G_{C_1} is weakly trained to put the target samples on the side of the unknown category and G_F is trained to deceive G_{C_1} by separating known and unknown target samples. G_F is trained with the ability to increase or decrease the probability of the classifier G_{C_1} for aligning target samples to known or unknown classes. The domain-invariant features are learned by following a minimax optimization procedure: the parameters θ_{G_F} of G_F are trained by maximizing the loss of G_{C_1}. On the other hand, we train the parameters of $\theta_{G_{C_1}}$ by minimizing the loss of G_{C_1}.

We quantify each target sample with a generated weight $W(x_j^t)$ (Section III-A2) and let the generator decide whether to recognize it as known or unknown. For example, when the value of $W(x_j^t)$ is high, it would be easier for generator to decide to decrease the probability $P(y = N + 1|x_j^t)$ lower than $W(x_j^t)$ to maximize the error of G_{C_1} and then align the target sample to source domain. This leads to the optimization problem as follows,

$$E_{G_{C_1}} = \frac{1}{n_s} \sum_{i=1}^{n_s} L_{G_{C_1}}(G_F(x_i^s), y_i^s) \qquad (2)$$

Here, $L_{G_{C_1}}$ is the cross-entropy loss function for minimizing the error of G_{C_1}. We use a binary cross-entropy loss for maximizing the error of G_{C_1} adversarially to create a boundary between known and unknown target samples, where we infuse our weight measures as follows,

$$E_{G_{C_1 adv}} = -\frac{1}{n_t} \sum_{j=1}^{n_t} W(x_j^t) \log(P(y = N + 1|x_j^t))$$
$$-\frac{1}{n_t} \sum_{j=1}^{n_t} (1 - W(x_j^t)) \log(1 - P(y = N + 1|x_j^t)) \qquad (3)$$

The minimax game between the generator and domain classifier in our model is equivalent to aligning weighted target samples towards known classes of the source domain or 'unknown' class. Our proposed open set domain adaptation

network can be trained by fulfilling a minimax optimization procedure as:

$$\theta_{G_{C_1}} = \underset{\theta_{G_{C_1}}}{\operatorname{argmin}} E_{G_{C_1}} + E_{G_{C_1 adv}}$$
$$\theta_{G_F} = \underset{\theta_{G_{C_1}}}{\operatorname{argmin}} E_{G_{C_1}} - E_{G_{C_1 adv}} \qquad (4)$$

Limitations of OSBP method The OSBP method is illustrated in Fig. 2b which has a similar adversarial model as discussed above The OSBP method [15] does not consider any domain level discriminative characteristics while aligning target samples towards known classes or rejecting them as unknown but rely only on the adversarial source classifier. OSBP method [15] sets an empirical fixed threshold value $T = 0.5$ to train the adversarial source classifier for unknown target samples, which may confuse the model during training by yielding similar chances of a sample being known and unknown at the same time and encourage negative transfers. For example, if the value of $W(x_j^t)$ is replaced by T in the binary cross-entropy loss optimization of Equation (3), the unknown category probability value worth $P(y = N + 1|x_j^t) \simeq 0.5$ for a target sample may produce similar amount of error for both known and unknown classes at the same time and confuse the generator G_F to decide whether to align the target sample towards known or unknown classes. This implies that OSBP method may encourage negative transfer.

To determine domain discriminative characteristics during training, we propose to integrate two additional classifiers such as supplementary source G_{C_2} and non-adversarial domain D classifiers in the adversarial model. The supplementary classifiers and the domain classifier G_{C_1} combinedly assist in identifying the 'unknownness' score ($W(x_j^t)$) of the target samples and encourage positive transfer. The output of the trained parameters of the supplementary non-adversarial domain classifier D gives the probability of a sample coming from source or target domain, and if it is from the target domain, then the output also holds its probability of belonging to the 'unknown' class. This constitutes our multi-classifier based weighting scheme as discussed in section III-A2.

The purpose of the weighting scheme is to generate distinguishable scores for known and unknown class target samples and hence reduce negative transfer. Generated weights are back-propagated for optimizing the generator and the domain classifier. The proposed weighting scheme assigns either high or low weights to the target samples depending on their similarity and dissimilarity to the source domain to assist the generator for maximizing the error of the domain classifier. For a detail explanation, please refer to section III-A2.

2) Weighting Scheme: The main challenge in our proposed method, as illustrated in Equations (2) and (3), is the way of quantifying the unknownness of each target samples $W(x_j^t)$. We aim to develop an unknownness measure $W(x_j^t)$ based on discriminative known-source label C_s information, domain information, and 'unknown' class probability information. We place a supplementary classifier G_{C_2} to predict the source class labels with a leaky-softmax function [17], which maintains the total probability less than 1. G_{C_2} converts features of G_F to

$|C_s|$-dimensional class probabilities through a leaky-softmax activation as,

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

where l is the logit vector. G_{C_2} is trained only on source samples, therefore, unlike source samples target samples will have smaller logits or uncertain predictions. The element-sum i.e. $d_1(x) = \sum_{k=1}^{|C_s|} G_{C_2}^k(G_F(x))$ of the leaky-softmax outputs for source domain samples will be high whereas, target samples will yield low outputs. We train G_{C_2} by a multiclass one-vs-rest binary loss for the $|C_s|$-class classification as,

$$E_{G_{C_2}} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{k=1}^{|C_s|} y_{i,k}^s \log G_{C_2}^k(G_F(x_i^s)) \\ + (1 - y_{i,k}^s) \log(1 - G_{C_2}^k(G_F(x_i^s))) \quad (6)$$

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 $G_{C_2}^k(G_F(x))$. Thus, $d_1(x_j^t)$ gives a combined measure of known source class label and domain information. For target samples, the higher the value of $d_1(x_j^t)$ is, the higher the chance that it comes from the shared label space C and is closer to the source domain. On the other hand, the smaller the value of $d_1(x_j^t)$ is the more probable that it comes from $\overline{C_t}$. This weight measure is still exposed to risk as to the value of $d_1(x_j^t)$ can be uncertain.

We further support the weight measure by the 'unknown' class probability value $P(y = N + 1|x_j^t)$ from the domain classifier G_{C_1} as follows,

$$d_2(x_j^t) = P(y = N + 1|x_j^t) d_1(x_j^t) \quad (7)$$

Now, $d_2(x_j^t)$ can be seen as the complete measure of the probability of target samples belonging to shared label space and unknown class as well. For example, it is highly likely that the target samples belonging to C will produce low 'unknown' probability $P(y = N+1|x_j^t)$ value and high $d_1(x_j^t)$ probability value, whereas, the target samples from $\overline{C_t}$ will produce a higher 'unknown' probability $P(y = N + 1|x_j^t)$ value and lower $d_1(x_j^t)$ value. Therefore, the output value of $d_2(x_j^t)$ will be different for known and unknown target samples making the job of G_F easier.

In particular, for the two deciding factors i.e. ($d_1(x_j^t)$ and $P(y = N + 1|x_j^t)$), the proposed weighting scheme needs to handle four cases of outcomes from the target samples: 1) The outputs of both deciding factors are low; 2) The output of $d_1(x_j^t)$ is low and $P(y = N+1|x_j^t)$ is high; 3) The output of $d_1(x_j^t)$ is high and $P(y = N + 1|x_j^t)$ is low; and 4) The outputs of both deciding factors are high. For Case 1, the output of $d_2(x_j^t)$ will be very low, indicating the generator to decide to increase the unknown probability to deceive the domain classifier. Though the output of the next two cases will be in the lower end, Case 3 will produce a higher $d_2(x_j^t)$ output than Case 2. In these two cases, the weighting scheme will assist the generator in deciding to increase the unknown probability for increasing the error of the domain classifier. For the last case, the output of $d_2(x_j^t)$ will be high which indicates

the generator needs to decrease the unknown probability to deceive the domain classifier.

Finally, being inspired by a prior work [49] that integrated the labeled information into the discriminator, we add a non-adversarial supplementary domain classifier D (i.e., the generator G_F does not deceive D adversarially) to integrate discriminative label and 'unknown' probability information. D assumes that the target samples belonging to the shared label space C are closer to the source domain samples than $\overline{C_t}$. As we want to train D to output high for source samples, we ignore the 'unknown' class probability value of source samples during optimization i.e., the encoded discriminative information with 'unknown' class probability ($d_2(x_j^t)$) for target samples and without 'unknown' class probability ($d_1(x_i^s)$) for source samples are integrated to D and train D to differentiate the source domain samples from the target domain samples to help generate weights. This observation leads to the training objective of D as follows,

$$E_D = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log d_1(x_i^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log(1 - d_2(x_j^t)) \quad (8)$$

Equation (9) indicates that the outputs of D are dependent on the output of the supplementary source classifier G_{C_2} and the unknown class probability $P(y = N+1|x_j^t)$ of the domain classifier G_{C_1} for target samples. This verifies that D is trained on the known-unknown label and domain information which will assist D to assign meaningful and distinguishable weights to target samples belonging to C and $\overline{C_t}$. Thus, we obtain a weight for each target samples as, $W(x_j^t) = D(x_j^t)$.

$$W(x_j^t) = D(x_j^t) \quad (9)$$

Considering all the above-discussed derivations, we present our final model with the proposed weight mechanism. We denote the parameters of G_{C_2} as $\theta_{G_{C_2}}$. The overall objectives of our proposed method are:

$$\theta_{G_{C_1}} = \underset{\theta_{G_{C_1}}}{\arg\min} E_{G_{C_1}} + E_{G_{C_{1adv}}}$$
$$\theta_{G_F} = \underset{\theta_{G_{C_1}}}{\arg\min} E_{G_{C_1}} - E_{G_{C_{1adv}}} \quad (10)$$
$$\theta_{G_{C_2}} = \underset{\theta_{G_{C_2}}}{\arg\min} E_{G_{C_2}} + E_D$$

For calculating the gradient of $E_{G_{C_{1adv}}}$ efficiently, we utilize a gradient reversal layer proposed by [7]. The job of the gradient reversal layer is to multiply the gradient by a certain negative constant while back-propagating. The gradient reversal phenomenon confirms that the feature distributions over the shared classes C of the source and target domains are made as indistinguishable as possible for the domain classifier G_{C_1}. Unlike prior work [15], 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.

B. Testing Phase

During the training phase, we fulfill our goal to transform G_{C_1} from source domain classifier to target domain classifier,

including the category 'unknown' by utilizing the supplementary source and domain classifiers. In the testing phase, we omit the supplementary classifiers and utilize only the trained feature generator G_F and the domain classifier G_{C_1} to classify test images correctly as shown in Fig. 2b.

IV. EXPERIMENTAL STUDIES

In this section, we describe the datasets, our evaluation details, and the results. We conduct experiments to evaluate our proposed approach with contemporary domain adaptation methods on four standard datasets and analyze the behavior of the proposed method in different domain adaptation settings.

A. Datasets

Office-31 [25] 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 domain adaptation. We have chosen the first 10 classes as C and the last 10 classes as 'unknown' samples in the target domain $\overline{C_t}$ for accomplishing six open set domain adaptation tasks: A→ W, D→ W, W→ D, A→ D, D→ A and W→ A.

VisDA2017 [50] poses a special domain adaptation setting by focusing on a simulation (rendered 3D images) to real-world domain adaptation setting. Game engines generate the samples of source domain while the target domain samples are actual images. This dataset comprises 12 categories. Inline with [15], 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 [51] 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 tot he target domain. For this dataset, we have designed twelve open set domain adaptation tasks: Ar→ Cl, Ar→ Pr, Ar→ Rw, Cl→ Ar, Cl→ Pr, Cl→ Rw, Pr→ Ar, Pr→ Cl, Pr→ Rw, Rw→ Ar, Rw→ Cl and Rw→ Pr.

ImageNet-Caltech is a combination of ImageNet-1K [52] consisting 1000 categories and Caltech-256 with 256 categories. In line with previous works [17], [18], 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 domain adaptation tasks I→ C and C→ 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 $|\overline{C_t}|$ as a unified 'unknown' class and the average per-class accuracy for all the $|C| + 1$ classes is the final result. We present the accuracy

measured over all the classes $|C| + 1$ as OS. We also show the average of per-class accuracy for the shared classes C (denoted as OS⋆) to compare the false-negative error rate of the proposed and contemporary domain adaptation works.

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 9. All experiments were executed in PyTorch. For a fair comparison, we have used the same backbone network and hyper-parameters to regenerate the results of contemporary domain adaptation works.

Compared Domain Adaptation Methods We have compared the performance of the proposed method with: **1)** Closed-set domain adaptation methods: Domain-Adversarial Neural Networks (DANN) [23] and Residual Transfer Networks (RTN) [13]; **2)** Partial domain adaptation methods: Importance Weighted Adversarial Nets (IWAN) [19] and Example Transfer Network (ETN) [17]; **3)** Open set domain adaptation methods: Unsupervised domain adaptation by back-propagation (BP) [7] with unknown source samples, Assign-and-Transform-Iteratively (ATI) [16] and Open Set domain adaptation by Back-Propagation (OSBP) [15]; **4)** Universal domain adaptation method: Universal Adaptation Network (UAN) [20]. Comparing our proposed method to these state-of-the-art methods in their respective domain adaptation settings will greater insight to our proposed methodś performance.

C. Classification results

The classification results on the twelve tasks of Office-Home, six tasks of Office-31, the task of VisDA and two 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). Our proposed method outperforms all the compared methods in terms of the average per-class accuracy. For contemporary partial, universal, and open set domain adaptation methods, the accuracy of the OS⋆ is less than that of OS for the majority of the tasks. This observation leads to the fact that the false-negative error rate is high which means a large number of known target images are classified as unknown. This is because domain adaptation methods such as BP (with unknown classes in the source domain) [7] and ATI [16] 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 false-negative transfer.

Although UAN [20] lags behind our performance, it has a less negative transfer rate compared to others for adapting sample-level transferability. IWAN and ETN also suffer from negative transfer effects for the majority of tasks. However, our proposed method yields better results for OS⋆ than OS which indicates a better separation of known and unknown target samples. The proposed method is prone to less negative transfer because of generated weights for target samples,

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

Approach	Ar→Cl		Ar→Pr		Ar→Rw		Cl→Ar		Cl→Pr		Cl→Rw		Pr→Ar		Pr→Cl		Pr→Rw		Rw→Ar		Rw→Cl		Rw→Pr		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
DANN [23] (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 [13] (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 [19] (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 [17] (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 [20] (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	83.4	82.5	63.2	63.0	79.1	78.1	76.8	76.6
BP [7] (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 [16] (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 [15] (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
Our(ResNet-50)	64.8	64.9	84.6	84.9	**88.1**	**88.2**	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)	**66.1**	**66.9**	**85.9**	**86.7**	87.6	87.9	**80.5**	**80.8**	**83.5**	**85.6**	**86.9**	**87.9**	**79.9**	**81.1**	**63.1**	**64.8**	**87.9**	**88.7**	**86.9**	**88.2**	**68.8**	**69.7**	**82.1**	**82.8**	**80.0**	**80.9**

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

Approach	A→W		D→W		W→D		A→D		D→A		W→A		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
DANN [23] (2016)	-	80.2	-	79.9	-	87.5	-	80.4	-	74.9	-	80.9	-	80.6
RTN [13] (2016)	-	88.1	-	89.8	-	84.8	-	73.1	-	85.1	-	84.7	-	84.2
IWAN [19] (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 [17] (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 [20] (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 [7] (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 [16] (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 [15] (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
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)	**90.2**	**90.8**	**97.9**	**98.8**	**98.6**	**98.8**	**89.5**	**89.7**	**91.0**	**91.7**	**86.8**	**87.6**	**92.4**	**92.9**

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

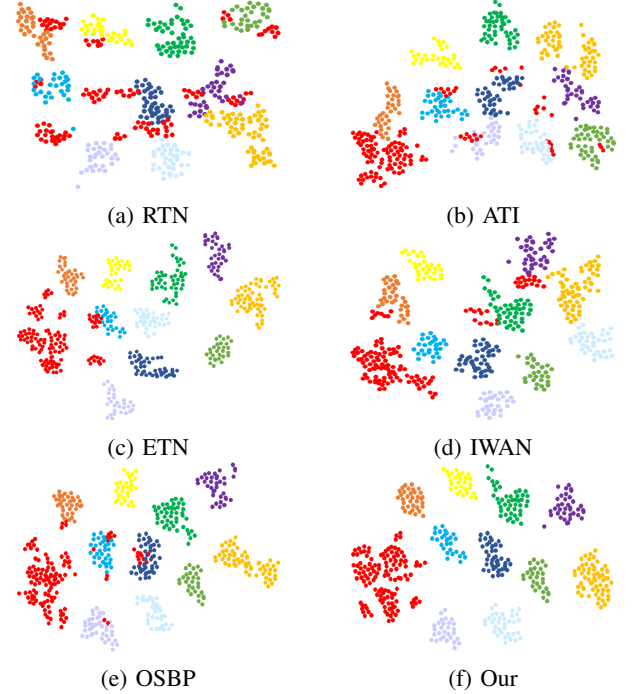| Approach	I→C		C→I		Avg	
	OS	OS*	OS	OS*	OS	OS*
DANN [23] (2016)	-	71.1	-	65.9	-	68.5
RTN [13] (2016)	-	71.9	-	66.2	-	69.1
IWAN [19] (2018)	74.1	72.9	68.7	65.3	71.4	69.1
ETN [17] (2019)	74.9	74.8	69.8	69.9		
UAN [20] (2019)	75.3	76.3	70.2	70.8	72.7	73.6
BP [7] (2015)	68.9	67.3	61.2	59.0	65.0	63.2
ATI [16] (2017)	71.6	65.9	67.4	65.1	69.5	65.5
OSBP [15] (2018)	63.1	63.4	54.8	53.6	58.9	58.5
Our(ResNet-50)	77.4	77.8	69.8	70.1	73.6	74.0
Our(ResNet-152)	**78.9**	**79.7**	**71.9**	**71.7**	**75.4**	**75.7**

(a) RTN (b) ATI (c) ETN (d) IWAN (e) OSBP (f) Our

Figure 3: Learned features of our proposed method for **A → W** task from **Office-31** dataset show a better separation of unknown samples (red dots) besides known classes than contemporary domain adaptation methods. We select 10 shared classes, 10 source domain private classes, and 10 target domain private classes for the task.

which are supported by the decision of both domain classifier G_{C_1} and supplementary source classifier G_{C_2}.

In particular, we have a crucial observation for OSBP [15], which follows the same domain adaptation setting like ours. Though OSBP demonstrates less negative transfer in Office-31, Office-Home, and ImageNet-Caltech tasks, its negative transfer rate increases at a great deal in the VisDA task. OSBP often confuses partially hidden known objects and the most prominent known object among multiple objects in an image by creating similar chances of being known and unknown. This leads to more negative transfers. On the other hand, the proposed method almost always initiates the positive transfer by assigning distinguishable weights to known and unknown target samples.

In Fig. 3, we plot the t-SNE [53] embeddings of the features learned by RTN [13], ATI [16], IWAN [19], ETN [17], OSBP [15] and proposed method on **A → W** task with 10 shared classes, 10 source domain private classes and 10 target domain private classes as per respective domain adaptation settings.

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

Approach	Accuracy (%)								
	bicycle	bus	car	motorcycle	train	truck	unknown	OS	OS*
DANN [23] (2016	32.4	51.6	65.1	71.3	85.1	23.1	-	-	52.1
RTN [13] (2016)	31.6	63.6	54.2	76.9	**87.3**	21.5	-	-	51.1
IWAN [19] (2018)	30.6	69.8	58.3	76.8	65.5	30.8	69.7	57.3	55.3
ETN [17] (2019)	31.6	66.8	61.7	77.8	70.8	30.8	70.7	58.6	56.6
UAN [20] (2019)	42.6	67.8	65.7	76.9	69.8	31.8	70.7	60.9	59.1
BP [7] (2015)	31.8	66.5	50.5	70.1	86.9	21.8	38.5	52.3	54.6
ATI [16] (2017)	33.6	51.6	64.2	78.1	85.3	22.5	42.5	54.8	52.6
OSBP [15] (2018)	35.6	59.8	48.3	76.8	55.5	29.8	**81.7**	55.4	50.9
Our(ResNet-50)	50.6	74.8	66.7	80.6	75.9	38.8	73.9	65.9	64.6
Our(ResNet-152)	**52.1**	**77.7**	**67.7**	**81.4**	80.8	**39.8**	75.5	**67.8**	**66.6**

The proposed method demonstrates significantly comprehensible and well-segregated clusters for all known and unknown classes than other domain adaptation 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 [15], 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 [13] and ATI [16] 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 [17] shows better clusters than IWAN [19] for initiating less false-negative transfers, it certainly lags behind our method.
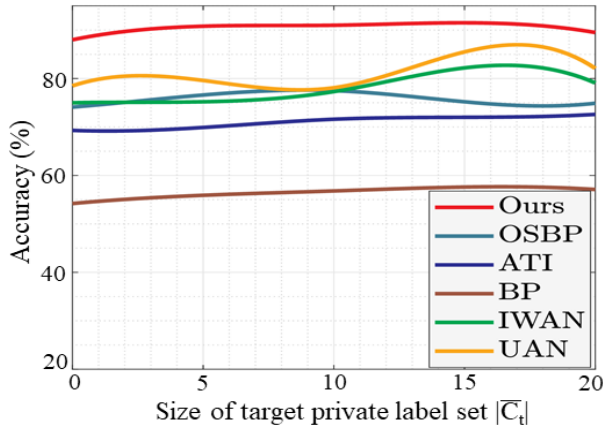

Figure 4: The proposed method performs better than contemporary domain adaptation methods with different domain adaptation settings for various $|\overline{C_t}|$. The performance of our proposed method increases significantly with an increase of $|\overline{C_t}|$. However, the rate of increment drops slightly after $|\overline{C_t}| = 15$. A large number of unknown classes in the target domain compared to the shared classes assist the source and supplementary source classifier in learning discriminative known class features which later helps to separate well known and unknown samples.
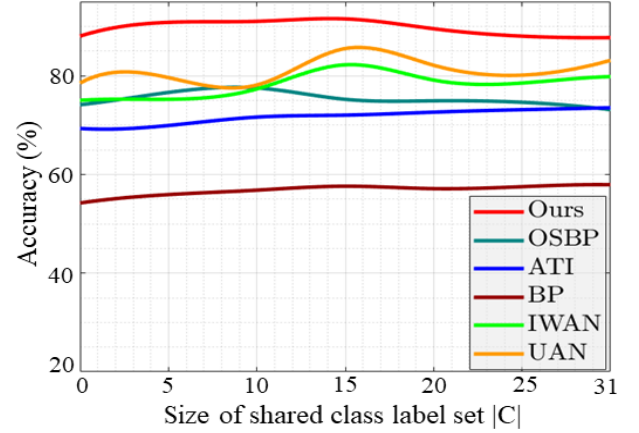

Figure 5: Our proposed method outperforms contemporary domain adaptation methods with different domain adaptation settings for different sizes of the shared label set. We observe that when $|C|$ reaches beyond 20, the performance of our method decreases slightly. This is because our method is designed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

D. Analysis on Various Open Set DA Settings

Varying size of $\overline{C_t}$ We compare the performance of our proposed method with other methods by varying the number of unknown samples in the target domain for D→ A task. Fig. 4 shows that our proposed method maintains a smooth increment of performance with no significant drop in-between transitions of varying $|\overline{C_t}|$ and outperforms all compared DA methods for all cases. This indicates a larger $|\overline{C_t}|$ assists both G_{C_1} and G_{C_1}, 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 any prior knowledge about the label sets like OSBP [15] and IWAN [19].

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→ 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-

9

Table V: Average 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
Our w/o P	63.3	86.1	93.1	96.2	84.2	86.5	82.1	88.0	75.3	65.4	70.4
Our w/o G_{C_2}	61.1	84.8	93.8	94.7	85.1	84.7	82.3	87.6	74.9	65.0	69.9
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

31 dataset has 31 categories when the shared label set C is beyond 20 the target private label set $|\overline{C_t}|$ 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 multi-classifier based weighting scheme to investigate deeper into its effectiveness.

Our w/o P is the variant of proposed method without integrating the 'unknown' class probability value $P(y = N+1|x_j^t)$ into the weighting scheme, i.e. in Equations (7) and (9). **Our w/o** G_{C_2} is the variant without integrating the domain-level information into the weighting scheme by omitting G_{C_2} and deploying D to depend only on the 'unknown' class probability value $P(y = N + 1|x)$ for source and target samples. To execute this variant, we need to omit Equation (5) and (6), omit $d_1(x)$ in Equation (7) and generalize Equation (7) for source samples as well, and replace $d_1(x_i^s)$ with $d_2(x_i^s)$ in Equation (9).

Table V presents the results for the variants of the proposed method, as mentioned above. The performance of both **Our w/o P** and **Our w/o** G_{C_2} lag behind the proposed method. This is because both the deciding factor $P(y = N+1|x_j^t)$ and $d_1(x_i^t)$ (output of G_{C_2}) are required for defining meaningful weights $W(x_j^t)$ to the target samples. We also observe that the variant without 'unknown' probability **Our w/o P** outperforms the other one, which indicates integrating the supplementary source classifier in the weighting scheme is more effective.

V. CONCLUSION

In this paper, we propose a multi-classifier based weighting scheme-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. Our proposed weighting scheme assists the feature generator and the domain classifier to distinguish known and unknown target samples by assigning identifiable scores to the target samples. Our weighting scheme does not harm the adversarial optimization by generating similar chances of a sample belonging to known and unknown classes at the same time. Thus, our proposed method is capable of enhancing positive target sample transfer by reducing the false-negative error rate.

A thorough experimental evaluation demonstrates that the proposed method consistently outperforms the existing domain adaptation methods. We further show through 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] 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.

[7] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning (ICML)*, 2015, pp. 1180–1189.

[8] 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.

[9] 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.

[10] 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.

[11] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *International Conference on Learning Representations (ICLR)*, 2016.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] P. P. Busto and J. Gall, "Open set domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 754–763.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," *arXiv preprint arXiv:1502.02791*, 2015.

[23] 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.

[24] 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.

[25] 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.

[26] X. Wang and J. Schneider, "Flexible transfer learning under support and model shift," in *Advances in Neural Information Processing Systems*, 2014, pp. 1898–1906.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," *arXiv preprint arXiv:1702.08811*, 2017.

[33] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 443–450.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[45] 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.

[46] 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.

[47] 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.

[48] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," *arXiv preprint arXiv:1707.07418*, 2017.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.