

Unknown Sample Discovery for Source Free Open Set Domain Adaptation

Chowdhury Sadman Jahan
Center for Imaging Science
Rochester Institute of Technology
Rochester, NY, USA
sj4654@rit.edu

Andreas Savakis
Department of Computer Engineering
Rochester Institute of Technology
Rochester, NY, USA
andreas.savakis@rit.edu

Abstract

Open Set Domain Adaptation (OSDA) aims to adapt a model trained on a source domain to a target domain that undergoes distribution shift and contains samples from novel classes outside the source domain. Test-time or source-free OSDA (SF-OSDA) techniques eliminate the need to access source domain samples, but current SF-OSDA methods utilize only the known classes in the target domain for adaptation, and require access to the entire target domain even during inference after adaptation, to distinguish between known and unknown samples. In this paper, we introduce Unknown Sample Discovery (USD) as an SF-OSDA method that utilizes a temporally ensembled teacher model to conduct known-unknown target sample separation and adapts the student model to the target domain over all classes using co-training and temporal consistency between the teacher and the student. USD promotes Jensen-Shannon distance (JSD) as an effective measure for known-unknown sample separation. Our teacher-student framework significantly reduces error accumulation resulting from imperfect known-unknown sample separation, while curriculum guidance helps to reliably learn the distinction between target known and target unknown subspaces. USD appends the target model with an unknown class node, thus readily classifying a target sample into any of the known or unknown classes in subsequent post-adaptation inference stages. Empirical results show that USD is superior to existing SF-OSDA methods and is competitive with current OSDA models that require both source and target domains during adaptation.

1. Introduction

The domain gap manifests when a model, trained on a source domain with annotated samples, is deployed in a target domain that has a distribution shift compared to the source domain. Unsupervised domain adaptation (UDA) considers a target domain with unlabeled data and aims to

mitigate the domain gap by aligning the source and target domain distributions. Most of the current UDA methods conduct domain adaptation by either minimizing the source-target distribution discrepancy [40, 48], or by adversarially aligning the feature spaces of the source and target data [7, 9]. However, such models need access to the source data during adaptation, and therefore cannot be applied to cases where the source domain is not available or when the source data is sensitive or confidential. To address these concerns, test-time or source-free domain adaptation (SFDA) [21, 46] was proposed, where domain adaptation to the target distribution takes place with a source-pretrained model using only the unlabelled target data.

While the vast majority of existing UDA literature deals with closed-set domain adaptation, where the target domain and source domain share the same classes, a more realistic scenario is open-set domain adaptation (OSDA) [23, 31] where the target domain contains samples belonging to novel classes that are absent in the source domain. In the OSDA setting, closed-set UDA would enforce alignment with unknown category mismatch, leading to negative transfer [6] and deteriorating performance. The majority of the existing OSDA methods [23, 31] utilize domain adversarial learning techniques to align the source domain with only the known classes in the target domain, leaving out the target-unknown classes. Such methods fail to properly learn the features for the unknown classes, and hence no clear decision boundary between the known classes and the unknown class in the target domain is realized. Some universal domain adaptation methods, i.e. UDA methods designed to work in both closed and open-set settings [16, 32], have attempted to conduct self-supervised learning (SSL) to discover latent target domain features without explicit distribution matching. However, such methods fail under large domain gaps. More recently, [13] proposed a three-way domain adversarial feature space alignment between the source domain and the known and the unknown target subdomains, thus segregating the known and unknown classes in the target domain.

In this work, we introduce **Unknown Sample Discovery (USD)** as a source-free OSDA (SF-OSDA) method that utilizes an ensemble-based pseudolabeling strategy for the target data, and generates known and unknown target subsets based on Jensen-Shannon distance (JSD) between the pseudolabels and the predictions from a teacher model. USD uses 2-component Gaussian Mixture Model (GMM) to model the target domain JSD, where the distribution with the lower mean JSD is considered to be of the known class samples and that with the greater mean JSD is taken as that consisting of unknown class samples. The known-unknown target subsets are used to adapt the student model. The student model is updated with gradient descent, while the teacher model is updated by exponential moving averages (EMA) of the teacher and student models. The teacher-student framework in USD helps to mitigate error accumulation induced from any possibly faulty known-unknown sample separation.

USD introduces an unknown class output node in the target model. The adapted target model infers new target samples in one of the known classes or the unknown class, without operating on the entire target dataset first to identify known and unknown samples. The main contributions of this work are as follows.

- We introduce USD as an SF-OSDA model that co-trains a dual-branch teacher-student framework to split the target domain into known and unknown class subsets.
- USD proposes the Jensen-Shannon distance between the target pseudolabels and teacher model predictions as an effective criterion for separating target samples in known and unknown classes.
- Co-training in USD, aided by weak-strong consistency between the teacher and student outputs, significantly mitigates error accumulation due to imperfect known-unknown separation and improves performance.
- USD generates reliable pseudolabels from the student model outputs on an ensemble of weak and strong target data augmentations.
- USD utilizes curriculum adaptation to progressively learn the known class feature space first, and the unknown class feature space later, thus enabling robust alignment of the entire target space with the source domain.
- Extensive experiments on 3 UDA datasets demonstrate the superiority of USD over existing SF-OSDA methods.

2. Related Works

2.1. Unsupervised domain adaptation

Domain gap originates from the distribution shift between the source domain where a deep network model is trained, and the target domain where the model is deployed [40]. This domain gap may be reduced by minimizing the maximum mean discrepancy (MMD) [24, 41], or the central mo-

ment discrepancy (CMD) [48] between the distributions in the source and target domains. Deep CORAL [36] mitigated domain shift by matching second-order distribution statistics. [7] introduced the Gradient Reversal Layer and made use of a domain discriminator to adversarially align the source and target distributions in a common feature space using a common feature encoder. [42] decoupled the feature extraction process by learning separate feature encoders for the two domains and aligned them adversarially to perform classification with a common classifier.

Generative adversarial networks (GANs) have been utilized to produce images in an intermediate domain between the source and target to facilitate easier and smoother adaptation [9]. Domain-wise global adversarial alignment in the absence of target annotations may lead to loss of class discrimination in the target embeddings. To align the domain-wise and class-wise distributions across the source and target data while maintaining target class feature discrimination, [18] simultaneously solved two complementary domain-specific and class-specific minimax objectives. [28] imposed a consistency constraint to non-adversarially align the labeled source prototypes and the pseudo-labeled target prototypes in the feature space.

2.2. Source free domain adaptation

UDA methods that adversarially align the embedding space [7, 9, 42] or minimize the source-target domain divergence [24, 41, 48] require access to both the source and target data during adaptation, rendering them unusable in situations where the source data is private or restricted. A semi-supervised UDA method involving a few source representatives or prototypes instead of the full source data was proposed in [4]. Distant supervision for SFDA [20] iteratively assigned pseudo-labels to the target data and used them to learn a domain invariant feature space and obtain the target class centroids. Liang *et al.* [21] introduced SHOT which adapts the source-pretrained feature encoder to the target domain via self-training with information maximization [14, 35] and self-supervised clustering for pseudolabeling, while transferring the source hypothesis (classifier model) to the target. To further refine the pseudolabels, [46] proposed to enforce neighborhood consistency regularization among the target samples. To generate compact target clusters, [47] considered minimizing the distance among K -nearest neighbors for each target sample and dispersing the rest by retrieving target features stored in a memory bank.

2.3. Open set domain adaptation

In addition to aligning the source and target subspaces, a critical step in OSDA is detecting target samples from novel or unknown categories that are absent in the source domain. [12] applied a simple class-wise confidence threshold to reject those samples with lower confidence as un-

known. [31] adversarially aligned the source domain and known target subdomain, where the unknown target samples were identified based on a preset threshold. Alignment for only the known classes however results in subpar performance in identifying the unknown samples. The adversarial alignment objective was modified in [23] with an instance weighting procedure, where higher weights were given to known target samples and lower weight to unknown samples. This somewhat smoothed the known-unknown distinction, but lower weights produced less contributions in the objective loss from the unknown samples, leading to suboptimal performance. A 3-way domain adversarial alignment between source, known target, and unknown target in the feature space was proposed in [13] such that the source and known target are aligned while the target-unknown gets segregated. [21] and [47] are SF-UDA methods that also conduct SF-OSDA by separating the known and unknown samples based on clustering the sample entropies into two clusters, and taking the cluster with lower mean entropy as the known subset.

3. Method

For unsupervised OSDA, we have n_s labeled samples $\{x_s^i, y_s^i\}_{i=1}^{n_s} \in \mathcal{X}_s, \mathcal{Y}_s$ belonging to the source domain \mathcal{D}_s , and n_t unlabeled samples $\{x_t^i\}_{i=1}^{n_t} \in \mathcal{X}_t$ belonging to the target domain \mathcal{D}_t . The task of SF-OSDA is to take the source model $f_s(\theta_s) : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ with model parameters θ_s trained on the C_s -multiclass source data $\{x_s^i, y_s^i\}_{i=1}^{n_s} \in \mathcal{X}_s, \mathcal{Y}_s$, and adapt it to $f_t(\theta_t) : \mathcal{X}_t \rightarrow \mathcal{Y}_t$ with model parameters θ_t that can map the $\{x_t^i\}_{i=1}^{n_t} \in \mathcal{X}_t$ to the C_t classes, where $C_t = C_s + 1$. The additional class in the target domain is a catch-all class for all samples in the target domain that do not belong to any of the classes in the source domain.

We follow [21] for Source model training follows [21] to ensure fair comparison with other source-free UDA models. The source model is trained by minimizing the standard cross entropy loss with label smoothing [27] as follows.

$$\mathcal{L}(f_s; \mathcal{X}_s, \mathcal{Y}_s) = -\mathbb{E}_{x_s \in \mathcal{X}_s, y_s \in \mathcal{Y}_s} \sum_{k=1}^{C_s} q_k^{ls} \log(\sigma_k(f_s(x_s))) \quad (1)$$

where $\sigma_k(a) = \frac{\exp(a_k)}{\sum_i \exp(a_i)}$ is the k -th element in its softmax output of a C_s -dimensional vector a , and q^{ls} is the one-hot encoded and smoothed C_s -dimensional vector for sample label y_s^i , such that $q_k^{ls} = (1 - \alpha)q_k + \alpha/C_s$, where q_k is 1 for the correct class and 0 for all other classes, and α is the smoothing factor set at 0.1.

The source model f_s consists of a feature extractor $g_s : \mathcal{X}_s \rightarrow \mathbb{R}^d$ and a C_s -class classifier $h_s : \mathbb{R}^d \rightarrow \mathbb{R}^{C_s}$, such that $f_s(x) = h_s(g_s(x))$. USD consists of a student target model $f_t^S(\theta_t^S)$ and a teacher target model $f_t^T(\theta_t^T)$. The feature extractors g_t^S and g_t^T , in the student and teacher net-

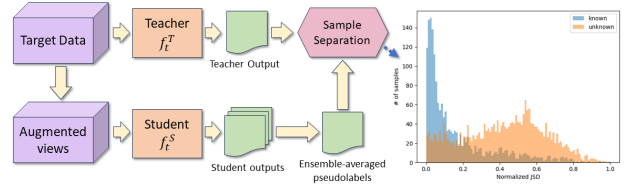


Figure 1. Pseudolabel generation for the target samples and known-unknown sample separation based on JSD

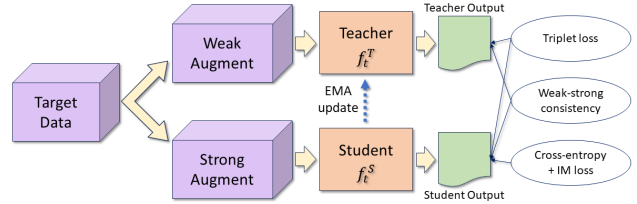


Figure 2. Adaptation process for USD using co-training. The student model receives pseudolabels for the target samples (see Figure 1) and is optimized using a combination of triplet, weak-strong consistency, information maximization (IM) and cross-entropy losses. The teacher model is updated via exponential moving averages (EMA) at the end of each epoch.

works respectively, are initialized with the source model feature extractor, i.e., $g_t^S = g_t^T = g_s$. To account for the novel class samples in the target domain, the source classifier h_s is expanded in the student and teacher models to include an additional trainable output node for the unknown class. The known class nodes in the target classifiers h_t^S and h_t^T , for the student and teacher respectively, are initialized with h_s , and remain frozen during adaptation. The unknown class nodes in h_t^S, h_t^T and the feature extractors g_t^S, g_t^T are adapted using only the unlabeled target samples.

3.1. Known-unknown sample separation

The first step for target adaptation is to reliably separate the known class samples and the novel class samples in the target data. This step is visually depicted in Figure 1. In order to generate pseudolabels \hat{y}_t , the target data undergoes $M = 6$ number of weak and strong augmentations (1 weak and 5 strong) based on *AutoAugment* [5] policy for ImageNet. The softmax output over C_s classes for each augmented view x_t^{iM} is taken from the student model f_t^S , and then averaged over the augmentations, as follows.

$$\hat{y}_t^i = \arg \max \frac{1}{M} \sum_1^M f_t^S(x_t^{iM}) \quad (2)$$

The index corresponding to the maximum averaged softmax output is taken as the hard pseudolabel \hat{y}_t^i for each target sample x_t^i . These pseudolabels are however only over the C_s known classes, and therefore the samples need to be

split into known class subset \mathcal{X}_t^K and unknown class subset \mathcal{X}_t^U . Existing SF-OSDA methods [21, 47] identify unknown class samples by utilizing the output entropy of the target data. Entropies for all samples are calculated at the beginning of each epoch and then normalized in the range of $[0, 1]$ by dividing each sample entropy by $\log C_s$. The normalized entropies are then clustered by 2-class k-means clustering. The cluster with the higher mean entropy or uncertainty is considered to be the one containing unknown samples, while the other cluster with lower mean entropy is taken as containing known class samples.

Sample separation is a critical component for noisy label learning (NLL) algorithms where clean and noisy samples are separated for robust supervised training of a model. Traditionally, NLL calculates the cross-entropy loss on the whole dataset and then uses low cross-entropy loss as the criterion to identify clean samples [1, 17, 19]. In USD, we conduct known-unknown sample separation for SF-OSDA based on JSD between the network outputs and their corresponding pseudolabels, which is calculated as follows.

$$JSD(\hat{y}_t^i, p_t^i) = \frac{1}{2}KL\left(\hat{y}_t^i, \frac{\hat{y}_t^i + p_t^i}{2}\right) + \frac{1}{2}KL\left(p_t^i, \frac{p_t^i + \hat{y}_t^i}{2}\right) \quad (3)$$

where, $KL(a, b)$ is the Kullback-Leibler divergence between a and b , and $p_t^i = \sigma(f_t^T(x_t^i))$ is the output softmax probability for target sample x_t^i from the teacher model f_t^T .

We consider the unknown class samples in the target domain as noisy samples [11] when predictions are made over only the known C_s classes. In comparison to entropy or cross-entropy loss, JSD is symmetric by design and ranges between 0 and 1. As shown in Figure 1, when plotted against the number of samples, JSD produces a bimodal histogram. A threshold could be applied directly on the JSD histogram for known-unknown separation, but such a threshold would depend on the location of the modes in the histogram, and hence inapplicable across all source-target pairs. We therefore model the JSD distribution with 2-component Gaussian Mixture Model (GMM) with equal priors, resulting in probabilities $[w_{t_L}^i, w_{t_H}^i]$ for each target sample x_t^i to belong to either of the two components. We consider the samples belonging to the distribution with the lower-mean Gaussian ($w_{t_L}^i > w_{t_H}^i$) as samples from one of the known classes, and consider those samples on the higher-mean Gaussian ($w_{t_L}^i < w_{t_H}^i$) as coming from the unknown target class.

Practically, we take the probability $w_{t_L}^i$ and in order to be conservative in our sample splitting, set a lower-bound/threshold δ_t to select the known sample subset \mathcal{X}_t^K . The remaining target samples are included in the unknown subset \mathcal{X}_t^U . The pseudolabels \hat{y}_t^i are updated accordingly, where the known subset retain their earlier assigned pseudolabel from among the C_s classes, and the unknown subset of target samples get the new unknown class pseudolabel $|C_t|$. It has to be noted that during adaptation, the teacher

network conducts the known-unknown sample separation at the beginning of each epoch, and the student network is adapted over the C_t classes with the target data.

3.2. Teacher-student co-training and regularization

USD simultaneously adapts the student and teacher target models, such that the student model parameters θ_t^S are updated based on the minibatch gradient descent, and the teacher network parameters θ_t^T are updated as temporally ensembled version of the student network [37] at the end of each epoch as follows.

$$\theta_{t_N}^T = m\theta_{t_{N-1}}^T + (1 - m)\theta_{t_N}^S \quad (4)$$

where, m is the momentum parameter for weight ensembling, and $N = 2, 3, \dots, E$ is the epoch number. Such co-training and cross-network sample splitting by the teacher for the student work to lessen error accumulation from imperfect known-unknown sample separation and stabilizes the adaptation process. As a means of training regularization, USD further maintains weak-strong temporal consistency between the teacher outputs and the student outputs by minimizing the following consistency loss.

$$\mathcal{L}_t^{con}(f_t^S, f_t^T; \mathcal{X}_t) = KL(p_t^{iS}, p_t^{iT}) = \sum_{k=1}^{C_t} p_t^{iT} \log\left(\frac{p_t^{iT}}{p_t^{iS}}\right) \quad (5)$$

where, $p_t^{iS} = \sigma(f_t^S(x_t^{iS}))$ is the softmax output from the student on a strongly augmented target sample x_t^{iS} , and $p_t^{iT} = \sigma(f_t^T(x_t^{iW}))$ is the softmax output from the teacher on the weakly augmented version x_t^{iW} of the same target instance. The strong and weak augmentations are done following the *AutoAugment* [5] ImageNet policy.

For contrastive learning in a label-free setting, it is exceedingly difficult to identify positive and negative instances for a target anchor. USD deftly utilizes the teacher-student framework and weak-strong augmentations, and applies a triplet loss [34] to effectively learn the decision boundary between known and unknown classes. The output $z_T^{ia} = [f_t^T(x_t^{iW})]^a$ of the teacher model on a weakly augmented known class sample is taken as the anchor, and the corresponding output $z_S^{i+} = [f_t^S(x_t^{iS})]^+$ on the strongly augmented version of the same sample from the student model is taken as the positive instance. The negative instance is the student model output $z_S^{i-} = [f_t^S(x_t^{iS})]^-$ on a randomly chosen unknown class sample. Cosine distance is taken as the distance metric, and is calculated as follows.

$$\mathbf{D}(z_1, z_2) = 1 - \frac{z_1 \cdot z_2}{\|z_1\|_2 \|z_2\|_2} \quad (6)$$

where z_1 and z_2 are any two network outputs. Triplet loss is in turn calculated as follows.

$$\mathcal{L}_t^{trip}(f_t^S, f_t^T; \mathcal{X}_t) = \max(\mathbf{D}(z_T^{ia}, z_S^{i+}) - \mathbf{D}(z_T^{ia}, z_S^{i-}) + 1, 0) \quad (7)$$

In addition, the student network is trained with the instance-weighted standard cross-entropy loss with label smoothing [27], as follows.

$$\mathcal{L}_t^{ce}(f_t^S; \mathcal{X}_t) = -\mathbb{E}_{x_t^i \in \mathcal{X}_t} \omega^i \sum_{k=1}^{C_t} \hat{y}_{t,k}^i \log(\sigma_k(f_t^S(x_t^{iS}))) \quad (8)$$

The instance weights ω^i are the probability w_{tL}^i for known target samples of belonging to the lower-mean JSD distribution, and w_{tH}^i for unknown target samples of belonging to the higher-mean JSD distribution, during the known-unknown sample separation. In order to promote adaptation to the known samples first and to progressively learn the unknown class feature space, USD utilizes cross-entropy loss under curriculum guidance, dictated by the curriculum factor γ_r as follows.

$$\mathcal{L}_t^{ce}(f_t^S; \mathcal{X}_t^K, \mathcal{X}_t^U) = \gamma_r \mathcal{L}_{tK}^{ce}(f_t^S; \mathcal{X}_t^K) + (1 - \gamma_r) \mathcal{L}_{tU}^{ce}(f_t^S; \mathcal{X}_t^U) \quad (9)$$

where $\gamma_r = \max(0.5, \gamma_{r-1}(1 - \beta \epsilon^{-\mathcal{L}_{tK_r}^{ce}/\mathcal{L}_{tK_{r-1}}^{ce}}))$ such that, β is a hyperparameter and r is the current iteration number. The ratio $\mathcal{L}_{tK_r}^{ce}/\mathcal{L}_{tK_{r-1}}^{ce}$ dictates the degree by which the curriculum factor decreases from the earlier $(r-1)$ -th iteration to the current r -th iteration. When loss \mathcal{L}_{tK}^{ce} on the known sample subset increases, γ marginally decreases to accommodate further adaptation on the known samples in the subsequent iterations. But if \mathcal{L}_{tK}^{ce} decreases by a large margin, γ decreases accordingly to progressively adapt to the unknown samples in the following iterations. Curriculum guidance balances the adaptation of the target model to the known and unknown subsets.

To encourage individually precise and globally diverse predictions, USD further minimizes the information maximization (IM) [21] loss as formulated in [38, 39].

$$\mathcal{L}_t^{ent}(f_t^S; \mathcal{X}_t^K) = -\mathbb{E}_{x_t^i \in \mathcal{X}_t^K} \sum_{k=1}^{C_t} \sigma_k(f_t^S(x_t^{iS})) \log(\sigma_k(f_t^S(x_t^{iS}))) \quad (10)$$

$$\mathcal{L}_t^{eqdiv}(f_t^S; \mathcal{X}_t^K) = \sum_{k=1}^{C_t} p_t^{iS} \log\left(\frac{p_t^{iS}}{\bar{p}_t^{iS}}\right) \quad (11)$$

$$\mathcal{L}_t^{IM}(f_t^S; \mathcal{X}_t^K) = \mathcal{L}_t^{ent}(f_t^S; \mathcal{X}_t^K) + \mathcal{L}_t^{eqdiv}(f_t^S; \mathcal{X}_t^K) \quad (12)$$

where $\bar{p}_t^{iS} = \mathbb{E}_{x_t^i \in \mathcal{X}_t^K} [\sigma(f_t^S(x_t^{iS}))]$ is the mean softmax output vector over known target samples in a minibatch. The overall objective function is therefore,

$$\mathcal{L}_t^{tot} = \mathcal{L}_t^{ce} + \mathcal{L}_t^{IM} + \zeta_1 \mathcal{L}_t^{trip} + \zeta_2 \mathcal{L}_t^{con} \quad (13)$$

where ζ_1 and ζ_2 are two hyperparameters.

A brief demonstration of the USD domain adaptation pipeline is presented in Algorithm 1.

Algorithm 1: Pseudocode for USD

Input: Source trained model f_s and n_t unlabeled target data samples $x_t^i \in \mathcal{X}_t$
Output: Target adapted student model f_t^S
Initialization: Teacher target model f_t^T and student target model f_t^S , are both initialized with parameters θ_s from f_s

- 1 **for** $epoch = 1$ to E **do**
- 2 Conduct $M = 6$ weak-strong augmentations and assign ensemble averaged pseudolabels \hat{y}_t^i using eq. (2)
- 3 Conduct known (\mathcal{X}_t^K) - unknown (\mathcal{X}_t^U) target sample separation using JSD between \hat{y}_t^i and teacher softmax output $p_t^i = \sigma(f_t^T(x_t^i))$
- 4 **for** $i = 1$ to n_t **do**
- 5 Optimize, for each minibatch, student model f_t^S with loss \mathcal{L}_{tot} using eq. (13) and get new student model parameters θ_t^S
- 6 **end**
- 7 Update teacher model f_t^T using new student model weights θ_t^S and current teacher model weights θ_t^T using eq. (4)
- 8 **end**

4. Experimental Setup

4.1. Datasets

We evaluate USD on three popular UDA benchmarks: Office-31 [30], Office-Home [43], and VisDA-C [29]. **Office-31** is a small-scale DA dataset with 3 distinct domains, Amazon (A), Webcam (W), and DSLR (D), containing images from 31 classes found in an office setting. Divided into 4 distinct domains Art (A), Clipart (C), Product (P), and Real-World (R), **Office-Home** is a medium-sized DA dataset which has images of 65 classes of objects found in contemporary office and home settings. The large-scale **VisDA-C** dataset contains images of 12 classes of items over 2 domains: Synthetic (S) and Real (R). Its source domain is composed of 152K synthetically rendered 3D images. The target domain consists of 55K real images taken from MS COCO dataset [22]. For OSDA, we follow the shared and target-private dataset splits done in [31].

4.2. Implementation details

For source training, we follow [21, 47] for fair comparison against existing SF-OSDA methods. The basic structure of the teacher and student models also follow that of [21, 47], that is, the feature extractor is a ResNet-50 [8], followed by a fully-connected (FC) bottleneck layer, a batch normalization layer [10], an FC classifier layer, and a weight normalization layer [33], respectively. The student target model is

trained with an SGD optimizer with momentum of 0.9 and weight decay of 10^{-3} . Due to the difference in the number of samples in each dataset, USD is adapted for 40 epochs on Office and for 20 epochs on Office-Home at $\beta = 0.01$, and for 5 epochs on VisDA-C at $\beta = 0.001$, at minibatch size of 64 samples in all cases. The threshold δ_t for known-unknown sample separation is set at 0.8, and the momentum parameter m for temporal ensembling is set according to the schedule in [45] with a maximum at 0.9995. Further, $\zeta_1 = 0.01$ and ζ_2 is gradually increased to 0.5 following [15]. All experiments were done on n A100 NVIDIA GPU.

4.3. Evaluation metrics

The mean-per-class accuracy **OS** over all known classes and the unified unknown class for all the target data may be considered as a metric for evaluating OSDA. However, such a metric is dominated by the accuracy on the known classes, as all the unknown samples are lumped into 1 unknown class [2]. A better metric is therefore to calculate the mean-per-class accuracy **OS*** over only the known classes, and the accuracy **UNK** for the unknown class, and then take the harmonic mean **HOS** of the two for fair evaluation over the known and the unknown classes. Mathematically, the metrics are formulated as follows.

$$OS^* = \frac{1}{|C_s|} \sum_{i=1}^{|C_s|} \frac{|x_t : x_t \in \mathcal{D}_t^i \cap \tilde{y}_t^i = i|}{|x_t : x_t \in \mathcal{D}_t^i|} \quad (14)$$

$$UNK = \frac{|x_t : x_t \in \mathcal{D}_t^{C_t} \cap \tilde{y}_t^i = C_t|}{|x_t : x_t \in \mathcal{D}_t^{C_t}|} \quad (15)$$

$$HOS = \frac{2 \times OS^* \times UNK}{OS^* + UNK} \quad (16)$$

Here, $\tilde{y}_t^i = \arg \max(\sigma(f_t^S(x_t^i)))$ is the prediction from the student model f_t^S and \mathcal{D}_t^i is the target domain data belonging to class i . In this work, we report OS*, UNK, and HOS for the evaluated adaptation tasks.

5. Results

5.1. Overall results

We compare USD to a number of existing UDA methods: closed-set UDA methods (1) DANN [7], (2) CDAN [25], open-set UDA methods (3) STA [23], (4) OSBP [31], (5) PGL [26], (6) OSLPP [44], and (7) UADAL [13]. These methods however are not source-free. We compare USD to open-set versions of SF-UDA methods SHOT [21] and AaD [47]. The open-set results for SHOT and AaD on Office-Home are provided in their respective publications. We generate results for Office-31 and VisDA-C using their publicly released code.

The results on Office-31 over all 6 domain pairs are presented in Table 1. USD outperforms SHOT and AaD by

$\sim 16\%$ and $\sim 3\%$, respectively in terms of mean HOS. Distinguishing between known and unknown class samples is crucial in OSDA, and USD strikes the best balance among the other SF-OSDA methods. SHOT clearly adapts primarily to the known classes without good adaptation on the unknown samples. AaD overcompensates in identifying unknown samples at the expense of correctly adapting to the known classes. USD performs equally well over both known and unknown classes, leading to higher HOS. USD also outperforms non-source-free methods STA and PGL, while being comparable to OSBP.

A comparative evaluation for USD against existing UDA methods on Office-Home is given in Table 2. USD outperforms SHOT and AaD by $\sim 20\%$ and $\sim 2\%$, respectively in terms of the average HOS over the 12 domain pairs. Similar to Office-31, SHOT adapts better to the known classes, but fails to competently identify unknown samples, while AaD performs worse on the known classes and better on the unknown samples. USD is more balanced across the known and unknown classes and also outperforms non-SF OSDA methods STA, OSBP and PGL.

Results on VisDA-C are given in the bottom right section in Table 2. SHOT severely suffers from negative transfer in the unknown class, while AaD fails to learn the target-known feature space. USD greatly outperforms SHOT and AaD, as well as the non-SF method OSBP, while being comparable to STA in terms of mean HOS.

5.2. Ablation study

A detailed ablation study was performed on the known-unknown sample selection criterion and on the modeling of the criterion distribution. The results of the ablation study on both Office-31 and VisDA-C are given in Table 3. USD uses JSD as the known-unknown sample splitting criterion, while entropy has been extensively used in existing OSDA methods (SHOT, AaD, UADAL etc.) for this purpose. In addition, cross-entropy (CE) loss is a popular criterion for separating clean-noisy samples for noisy label learning (NLL) algorithms [1, 17, 19]. We evaluate all three criteria to find the best performing one. The criterion distribution can be modelled by either Gaussian Mixture Model (GMM) or a Beta Mixture Model (BMM). UADAL models sample entropy distribution using BMM to distinguish between known and unknown samples. Our results in Table 3 show that modeling the distribution of the JSD with a GMM outperforms all of the other combinations for unknown sample discovery.

The effect of the JSD threshold δ_t for known-unknown separation on the final HOS is shown in Figure 3. The performance is relatively uniform, which suggests robustness of adaptation to the hyperparameter δ_t . Nonetheless, if the threshold is set too high (eg. 0.9), too few samples may be denoted as known samples, leading to inferior performance.

Method	SF	A → D			A → W			D → A			D → W			W → A			W → D			Avg.		
		OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
DANN[7]	X	90.8	59.2	71.5	87.4	55.7	68.1	72.9	74.5	73.7	99.3	77.0	86.7	72.1	73.1	72.6	100.0	70.2	82.5	87.1	68.3	75.9
CDAN[25]	X	92.2	52.4	66.8	90.3	50.7	64.9	74.9	70.6	72.7	99.6	73.2	84.3	72.8	69.3	71.0	100.0	67.3	80.5	88.3	63.9	73.4
STA[23]	X	91.0	63.9	75.0	86.7	67.6	75.9	83.1	65.9	73.2	94.1	55.5	69.8	66.2	68.0	66.1	84.9	67.8	75.2	84.3	64.8	72.5
OSBP[31]	X	90.5	75.5	82.4	86.8	79.2	82.7	76.1	72.3	75.1	97.7	96.7	97.2	73.0	74.4	73.7	99.1	84.2	91.1	87.2	80.4	83.7
PGL[26]	X	82.1	65.4	72.8	82.7	67.9	74.6	80.6	61.2	69.5	87.5	68.1	76.5	80.8	61.8	70.1	82.8	64.0	72.2	82.7	64.7	72.6
OSLPP[44]	X	92.6	90.4	91.5	89.5	88.4	89.0	82.1	76.6	79.3	96.9	88.0	92.3	78.9	78.5	78.7	95.8	91.5	93.6	89.3	85.6	87.4
UADAL[13]	X	85.1	87.0	86.0	84.3	94.5	89.1	73.3	87.3	79.7	99.3	96.3	97.8	67.4	88.4	76.5	99.5	99.4	99.5	84.8	92.1	88.1
SHOT*[21]	✓	94.0	46.3	62.0	95.6	42.3	58.7	83.3	39.1	53.3	100.0	75.7	86.1	82.7	46.6	59.6	100.0	69.7	82.1	92.6	53.3	67.0
AaD*[47]	✓	73.0	84.6	78.3	63.5	89.5	74.3	63.6	88.9	74.2	78.0	98.5	87.0	61.9	88.9	73.0	94.6	96.8	95.7	72.4	91.2	80.4
USD (Ours)	✓	90.7	73.4	81.2	82.8	72.7	77.9	65.7	84.4	73.9	97.9	96.6	97.3	64.6	86.7	74.0	98.0	92.6	95.2	83.3	84.4	83.3

Table 1. Evaluation of USD on Office-31 dataset. * are results computed for the methods using publicly released code.

Method	SF	Office-Home																		VisDA-C			
		A → C			A → P			A → R			C → A			C → P			C → R						P → A
		OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*
DANN[7]	X	37.1	82.7	51.2	60.0	71.3	65.2	75.1	67.3	71.0	43.8	84.3	57.6	50.1	77.6	60.9	61.1	73.5	66.7	42.4	83.9	56.3	
CDAN[25]	X	39.7	78.9	52.9	61.7	68.8	65.1	75.2	66.7	70.7	44.9	82.8	58.2	51.6	76.8	61.7	61.5	73.7	67.1	45.8	81.2	58.6	
STA[23]	X	46.0	72.3	55.8	68.0	48.4	54.0	78.6	60.4	68.3	51.4	65.0	57.4	61.8	59.1	60.4	67.0	66.7	66.8	54.2	72.4	61.9	
OSBP[31]	X	50.2	61.1	55.1	71.8	59.8	65.2	79.3	67.5	72.9	59.4	70.3	64.3	67.0	62.7	64.7	72.0	69.2	70.6	59.1	68.1	63.2	
PGL[26]	X	63.3	19.1	29.3	78.9	32.1	45.6	87.7	40.9	55.8	85.9	5.3	10.0	73.9	24.5	36.8	70.2	33.8	45.6	73.7	34.7	47.2	
OSLPP[44]	X	55.9	67.1	61.0	72.5	73.1	72.8	80.1	69.4	74.3	49.6	79.0	60.9	61.6	73.3	66.9	67.2	73.9	70.4	54.6	76.2	63.6	
UADAL[13]	X	54.9	74.7	63.2	69.1	72.5	70.8	81.3	73.7	77.4	53.5	80.5	64.2	62.1	78.8	69.5	69.1	78.3	73.4	50.5	83.7	63.0	
SHOT[21]	✓	67.0	28.0	39.5	81.8	26.3	39.8	87.5	32.1	47.0	66.8	46.2	54.6	77.5	27.2	40.2	80.0	25.9	39.1	66.3	51.1	57.7	
AaD[47]	✓	50.7	66.4	57.6	64.6	69.4	66.9	73.1	66.9	69.9	48.2	81.1	60.5	59.5	63.5	61.4	67.4	68.3	67.8	47.3	82.4	60.1	
USD (Ours)	✓	53.3	71.5	61.1	65.7	74.9	70	73.3	79.5	76.3	52.2	70.8	60.1	62.4	68.4	65.2	69.3	68.6	68.9	54.3	73.8	62.6	

Method	SF	Office-Home															VisDA-C						
		P → C			P → R			R → A			R → C			R → P						Avg.			
		OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*
DANN[7]	X	30.1	86.3	44.6	67.7	72.0	69.8	56.8	77.1	65.4	37.1	80.9	50.9	69.6	67.2	68.4	52.6	77.1	60.7	52.1	-	-	
CDAN[25]	X	33.1	82.4	47.2	69.8	69.7	69.7	59.8	73.6	66.0	40.3	75.8	52.7	70.9	64.6	67.6	54.5	74.6	61.4	-	-	-	
STA[23]	X	44.2	67.1	53.2	76.2	64.3	69.5	67.5	66.7	67.1	49.9	61.1	54.5	77.1	55.4	64.5	61.8	63.3	61.1	62.4	82.4	71.0	
OSBP[31]	X	44.5	66.3	53.2	76.2	71.7	73.9	66.1	67.3	66.7	48.0	63.0	54.5	76.3	68.6	72.3	64.1	66.3	64.7	50.9	81.7	62.7	
PGL[26]	X	59.2	38.4	46.6	84.8	27.6	41.6	81.5	6.1	11.4	68.8	0.0	0.0	84.8	38.0	52.5	76.1	25.0	35.2	-	-	-	
OSLPP[44]	X	53.1	67.1	59.3	77.0	71.2	74.0	60.8	75.0	67.2	54.4	64.3	59.0	78.4	70.8	74.4	63.8	71.7	67.0	-	-	-	
UADAL[13]	X	43.4	81.5	56.6	71.6	83.1	76.9	66.7	78.6	72.1	51.1	74.5	60.6	77.4	76.2	76.8	62.6	78.0	68.7	-	-	-	
SHOT[21]	✓	59.3	31.0	40.8	85.8	31.6	46.2	73.5	50.6	59.9	65.3	28.9	40.1	84.4	28.2	42.3	74.6	33.9	45.6	57.5*	12.1*	20.1*	
AaD[47]	✓	45.4	72.8	55.9	68.4	72.8	70.6	54.5	79.0	64.6	49.0	69.6	57.5	69.7	70.6	70.1	58.2	71.9	63.6	32.0*	62.9*	42.4*	
USD (Ours)	✓	47.3	69.6	56.3	70	74.5	72.2	64.6	71.3	67.8	53.8	65.5	59.1	73.3	69.1	71.1	61.6	71.5	65.9	57.8	86.7	69.4	

Table 2. Evaluation of USD on Office-Home and VisDA-C datasets. * are results computed for the methods using publicly released code.

Table 4 shows the impact of different components of our objective function and the effects of our teacher-student co-training scheme on the final adaptation performance for Office-Home. It is evident that each of our losses (\mathcal{L}_t^{trip} , \mathcal{L}_t^{con} , \mathcal{L}_t^{IM}) contributes to the adaptation, and leaving out any one of them hurts performance. We observe that curriculum guidance considerably benefits adaptation and the final average HOS increases by $> 1.5\%$ (from 64.2% to 65.9%) when such guidance is included. Notably, without curriculum, adaptation to the known classes is impacted drastically (OS* falls by $\sim 6\%$), signalling that progressively learning the known class subspace first and then the unknown class subspace later is the superior strategy.

The final row in Table 4 presents results in the absence of the teacher network, where the student network conducts the known-unknown sample separation for itself. Both the weakly and strongly augmented samples are fed through the student network, and losses \mathcal{L}_t^{trip} , \mathcal{L}_t^{con} are calculated over the student model outputs between the weak and strong aug-

mentations. Empirical results clearly show that co-training in a teacher-student framework is pivotal for mitigating the effect of any imperfect known-unknown separation and average HOS over the 12 domain pairs in Office-Home decreases by $\sim 5\%$ when the teacher network is removed. As seen in Figure 4, in the absence of co-training, the student model adapts faster, but its performance drops from its peak during the course of adaptation due to error accumulation. In contrast, adaptation with co-training is slightly slower but maintains its peak performance.

The effect of the pseudolabeling scheme on the adaptation performance for Office-Home is shown in Table 5. SHOT and AaD use a self-supervised clustering process built on DeepCluster [3] to get pseudolabels for the known samples. But in open set settings, the unknown samples can drift the known class centroids, leading to faulty clusters. Our multi-view augmentation ensembled pseudolabeling strategy outperforms both pseudolabeling from clustering or direct student predictions.

Separation criterion	Distribution modeling	Office																		VisDA-C					
		A → D			A → W			D → A			D → W			W → A			W → D			Avg.					
		OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
JSD	GMM	89.4	70.2	78.6	82.7	73.0	77.6	66.4	85.2	74.6	97.5	97.0	97.2	68.3	85.4	75.9	98.0	93.6	95.8	83.7	84.1	83.3	57.8	86.7	69.4
Entropy	GMM	88.9	70.2	78.4	83.3	74.5	78.6	65.3	90.5	75.9	97.9	93.3	95.5	60.2	88.5	71.7	98.0	93.1	95.5	82.3	85.0	82.6	57.1	85.4	68.4
CE	GMM	90.7	68.6	78.1	90.0	61.8	73.3	69.6	81.0	74.9	98.2	93.3	95.6	68.5	86.0	76.2	98.0	90.4	94.1	85.8	80.2	82.0	67.3	45.5	54.3
JSD	BMM	91.4	53.7	67.7	93.6	53.2	67.8	77.7	72.3	74.9	100.0	82.4	90.3	77.1	72.5	74.8	100.0	71.3	83.2	90.0	67.6	76.5	67.6	58.3	62.6
Entropy	BMM	90.2	60.1	72.1	87.2	78.3	82.5	66.3	88.5	75.8	89.5	92.1	90.8	60.8	87.1	71.6	100.0	88.8	94.1	82.3	82.5	81.2	42.3	83.4	56.1
CE	BMM	96.0	25.0	39.7	93.6	37.8	53.9	81.0	61.2	69.7	100.0	63.3	77.5	78.4	68.0	72.8	100.0	71.3	83.2	91.5	54.4	66.1	68.6	24.0	35.5

Table 3. Evaluation of separation criterion and distribution modeling for known-unknown sample separation in USD on Office dataset.

Method	A → C			A → P			A → R			C → A			C → P			C → R			Avg.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS		OS*	UNK
USD (full)	53.3	71.5	61.1	65.7	74.9	70.0	73.3	79.5	76.3	52.2	70.8	60.1	62.4	68.4	65.2	69.3	68.6	68.9	61.6	71.5	65.9
USD w/o L_t^{trip}	52.9	69.9	60.2	66.4	75.1	70.4	73.6	78.9	76.2	52.0	70.0	59.3	62.3	68.5	65.2	68.0	67.8	67.9	61.0	71.3	65.4
USD w/o L_t^{con}	50.5	75.6	60.6	63.3	77.7	69.8	69.6	83.1	75.8	49.4	74.4	59.3	57.9	73.8	64.9	64.3	72.9	68.3	58.0	75.4	65.3
USD w/o L_t^{IM}	50.1	74.7	59.9	64.6	73.5	68.7	73.5	77.7	75.5	51.2	67.9	58.4	60.2	68.3	64.0	66.7	69.0	67.9	59.8	71.3	64.6
USD w/o curriculum	47.5	77.1	58.8	60.8	79.4	68.9	69.3	82.5	75.3	44.7	79.1	57.1	57.8	74.6	65.2	62.2	73.8	67.5	55.7	77.0	64.2
USD w/o co-training	44.0	80.4	56.8	58.5	78.4	67.0	64.1	78.2	70.5	43.4	72.3	54.3	50.5	71.0	59.0	51.4	76.1	61.4	51.0	76.7	61.0

Method	P → A			P → C			P → R			R → A			R → C			R → P			Avg.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS		OS*	UNK
USD (full)	54.3	73.8	62.6	47.3	69.6	56.3	70	74.5	72.2	64.6	71.3	67.8	53.8	65.5	59.1	73.3	69.1	71.1	61.6	71.5	65.9
USD w/o L_t^{trip}	51.2	75.9	61.1	47.6	70.4	56.8	69.5	74.2	71.8	63.9	69.9	66.8	51.4	66.9	58.2	73.5	67.6	70.4	61.0	71.3	65.4
USD w/o L_t^{con}	49.4	78.0	60.5	44.9	71.8	55.3	66.0	78.4	71.7	60.3	75.3	67.0	50.1	70.1	58.4	70.6	73.9	72.2	58.0	75.4	65.3
USD w/o L_t^{IM}	50.6	75.4	60.6	45.5	68.1	54.5	68.7	74.1	71.3	63.2	72.8	67.7	49.7	66.5	56.9	73.1	67.3	70.1	59.8	71.3	64.6
USD w/o curriculum	46.1	80.7	58.6	40.4	74.5	52.4	64.7	78.3	70.9	57.9	77.7	66.4	48.0	73.3	58.0	69.0	73.4	71.1	55.7	77.0	64.2
USD w/o co-training	48.3	78.9	59.9	38.5	71.8	50.1	51.6	79.2	62.5	53.9	76.5	63.2	46.6	77.6	58.2	60.7	80.5	69.2	51.0	76.7	61.0

Table 4. Ablation study on the objective function, and co-training for USD on Office-Home dataset.

Pseudolabel	A → C			A → P			A → R			C → A			C → P			C → R			Avg.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS		OS*	UNK
Ensemble	53.3	71.5	61.1	65.7	74.9	70.0	73.3	79.5	76.3	52.2	70.8	60.1	62.4	68.4	65.2	69.3	68.6	68.9	61.6	71.5	65.9
Clustering	50.8	73.5	60.1	67.0	73.2	69.9	74.8	75.8	75.3	54.3	67.0	60.0	61.5	66.9	64.1	67.2	66.1	66.7	61.2	70.5	65.2
Student Predictions	50.7	74.1	60.2	65.9	73.6	69.5	74.7	78.5	76.5	51.3	68.2	58.6	61.7	67.3	64.4	67.2	69.5	68.3	60.6	71.6	65.3

Pseudolabel	P → A			P → C			P → R			R → A			R → C			R → P			Avg.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS		OS*	UNK
Ensemble	54.3	73.8	62.6	47.3	69.6	56.3	70.0	74.5	72.2	64.6	71.3	67.8	53.8	65.5	59.1	73.3	69.1	71.1	61.6	71.5	65.9
Clustering	53.4	73.6	61.9	48.6	69.0	57.1	71.0	71.8	71.4	63.4	72.2	67.5	52.2	68.1	59.1	70.0	68.3	69.2	61.2	70.5	65.2
Student Predictions	53.0	73.2	61.5	46.6	71.3	56.4	68.9	73.8	71.3	61.8	73.0	66.9	53.3	65.8	58.9	72.7	70.6	71.6	60.6	71.6	65.3

Table 5. Ablation study on the pseudolabeling scheme for USD on Office-Home dataset.

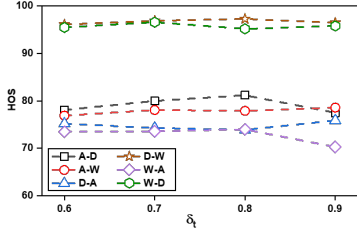


Figure 3. Impact of JSD threshold δ_t on HOS for Office dataset.

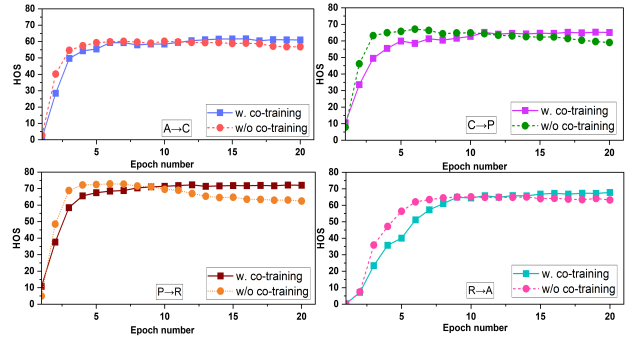


Figure 4. Impact of co-training on reducing error accumulation during adaptation on Office-Home dataset.

6. Conclusion

We present Unknown Sample Discovery as a teacher-student co-training framework that conducts SF-OSDA by splitting the target data into known and unknown subsets based on the JSD criterion modeled with a 2-component Gaussian mixture model. Co-training regularization greatly mitigates error accumulation, while curriculum guidance progressively adapts the target model to effectively learn both the known and unknown target feature spaces. Empirically, USD outperforms existing SF-OSDA methods and is comparable to non-source-free OSD techniques.

Acknowledgements

This research was partly supported by AFOSR grants FA9550-20-1-0039 and FA9550-22-P-0009 with Intelligent Fusion Technology, and the NYSTAR Center of Excellence in Data Science through the University of Rochester. The authors would like to thank RIT Research Computing for making computing resources available.

References

- [1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR, 2019. 4, 6
- [2] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *European conference on computer vision*, pages 422–438. Springer, 2020. 6
- [3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 7
- [4] Boris Chidlovskii, Stephane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 451–460, 2016. 2
- [5] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 3, 4
- [6] Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE transactions on neural networks and learning systems*, 32(10):4309–4322, 2020. 1
- [7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 2, 6, 7
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [9] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018. 1, 2
- [10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 5
- [11] Chowdhury Sadman Jahan and Andreas Savakis. Curriculum guided domain adaptation in the dark. *arXiv preprint arXiv:2308.00956*, 2023. 4
- [12] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part III 13*, pages 393–409. Springer, 2014. 2
- [13] JoonHo Jang, Byeonghu Na, Dong Hyeok Shin, Mingi Ji, Kyungwoo Song, and Il-Chul Moon. Unknown-aware domain adversarial learning for open-set domain adaptation. *Advances in Neural Information Processing Systems*, 35:16755–16767, 2022. 1, 3, 6, 7
- [14] Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in neural information processing systems*, 23, 2010. 2
- [15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 6
- [16] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9757–9766, 2021. 1
- [17] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 4, 6
- [18] Shuang Li, Chi Harold Liu, Binhui Xie, Limin Su, Zhengming Ding, and Gao Huang. Joint adversarial domain adaptation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 729–737, 2019. 2
- [19] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 316–325, 2022. 4, 6
- [20] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2975–2984, 2019. 2
- [21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020. 1, 2, 3, 4, 5, 6, 7
- [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5
- [23] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2927–2936, 2019. 1, 3, 6, 7
- [24] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2
- [25] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 6, 7

- [26] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. In *International Conference on Machine Learning*, pages 6468–6478. PMLR, 2020. 6, 7
- [27] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 3, 5
- [28] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2239–2247, 2019. 2
- [29] Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2021–2026, 2018. 5
- [30] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV 11*, pages 213–226. Springer, 2010. 5
- [31] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 153–168, 2018. 1, 3, 5, 6, 7
- [32] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *Advances in neural information processing systems*, 33:16282–16292, 2020. 1
- [33] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016. 5
- [34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 4
- [35] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1275–1282, 2012. 2
- [36] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016. 2
- [37] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 4
- [38] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. Unsupervised continual learning for gradually varying domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3740–3750, 2022. 5
- [39] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. Continual unsupervised domain adaptation in data-constrained environments. *IEEE Transactions on Artificial Intelligence*, 2023. 5
- [40] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 1, 2
- [41] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2
- [42] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 2
- [43] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 5
- [44] Qian Wang, Fanlin Meng, and Toby P Breckon. Progressively select and reject pseudo-labelled samples for open-set domain adaptation. *arXiv preprint arXiv:2110.12635*, 2021. 6, 7
- [45] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14383–14392, 2021. 6
- [46] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 1, 2
- [47] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, and Joost van de weijer. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems*, 2022. 2, 3, 4, 5, 6, 7
- [48] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *International Conference on Learning Representations*, 2017. 1, 2