Domain Consensus Clustering for Universal Domain Adaptation

Guangrui Li[1], Guoliang Kang[2†], Yi Zhu[3], Yunchao Wei[1], Yi Yang[1]
[1] ReLER Lab, AAII, University of Technology Sydney
[2] Carnegie Mellon University, [3] Amazon Web Services

Abstract

In this paper, we investigate Universal Domain Adaptation (UniDA) problem, which aims to transfer the knowledge from source to target under unaligned label space. The main challenge of UniDA lies in how to separate common classes (i.e., classes shared across domains), from private classes (i.e., classes only exist in one domain). Previous works treat the private samples in the target as one generic class but ignore their intrinsic structure. Consequently, the resulting representations are not compact enough in the latent space and can be easily confused with common samples. To better exploit the intrinsic structure of the target domain, we propose Domain Consensus Clustering (DCC), which exploits the domain consensus knowledge to discover discriminative clusters on both common samples and private ones. Specifically, we draw the domain consensus knowledge from two aspects to facilitate the clustering and the private class discovery, i.e., the semantic-level consensus, which identifies the cycle-consistent clusters as the common classes, and the sample-level consensus, which utilizes the cross-domain classification agreement to determine the number of clusters and discover the private classes. Based on DCC, we are able to separate the private classes from the common ones, and differentiate the private classes themselves. Finally, we apply a class-aware alignment technique on identified common samples to minimize the distribution shift, and a prototypical regularizer to inspire discriminative target clusters. Experiments on four benchmarks demonstrate DCC significantly outperforms previous state-of-the-arts.

1. Introduction

Deep convolutional neural networks have achieved significant progress in many fields, such as image classification [55, 24], semantic segmentation [7, 8], *etc*. However, as a data-driven technique, the severe reliance on annotated in-domain data greatly limits its application to cross-domain tasks. As a feasible solution, unsupervised domain adaptation (UDA) [45] tries to solve this by transferring the knowledge from an annotated domain to an unlabeled domain, and has achieved significant progress in multiple tasks [33, 34, 27, 29, 35, 37]. Despite UDA's achievement, most UDA solutions assume that two domains share identical label set, which is hard to satisfy in real-world scenarios.

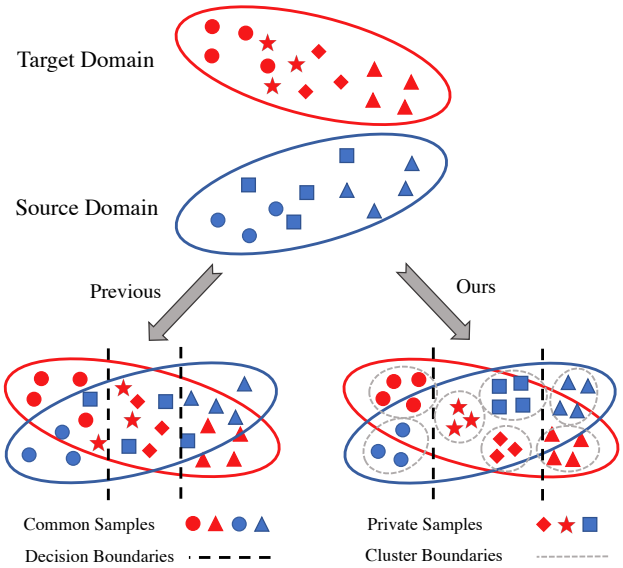


Figure 1. A comparison between previous methods and ours. Previous methods simply treat private samples as one general class and ignore its intrinsic data structure. Our approach aims to better exploit the diverse distribution of private samples via forming discriminative clusters on both common samples and private samples.

In light of this, several works considering the unaligned label set have been proposed: open set domain adaptation, partial domain adaptation, and universal domain adaptation. Open set domain adaptation (OSDA) [54] assumes the target domain possesses private classes that are unknown to the source domain. Analogously, partial domain adaptation (PDA) [4] describes a setting where only the source domain holds private classes. However, both OSDA and PDA still require prior knowledge where the private classes lie in. As a result, they are limited to one scenario and fail to generalize to other scenarios. For example, an OSDA solution

†Corresponding author.
Code available at: https://git.io/JY86C

would fail in the PDA scenario as it only seeks private samples in the target domain. To solve this, [62] takes a step further to propose a more general yet practical setting, universal domain adaptation (UniDA), which allows both domains to own private classes.

The main challenge of transferring over unaligned label space is how to effectively separate common samples from private samples in both domains. To achieve this goal, many efforts have been devoted to performing common sample discovery from different perspectives, such as designing new criteria [4, 3, 62, 17, 53] or introducing extra discriminator [63, 5, 38, 9]. However, previous practices mainly focus on identifying common samples but treat private samples as a whole, *i.e*., *unknown* class (Bottom left in Fig. 1). Despite making progress, the *intrinsic* structure (*i.e*., the variations within each semantic class and the relationships between different semantic classes) of the private samples is not fully exploited. As the private samples in nature belong to distinct semantic classes, treating them as one general class is arguably sub-optimal, which further induces lower compactness and less discriminative target representations.

In this paper, we aim to better exploit the intrinsic structure of the target domain via mining both common classes and individual private classes. We propose Domain Consensus Clustering (DCC), which utilizes the domain consensus knowledge to form discriminative clusters on both common samples and private samples (Bottom right in Fig. 1). Specifically, we mine the domain consensus knowledge from two aspects, *i.e*., semantic-level and sample-level, and integrate them into two consecutive steps. Firstly, we leverage Cycle-Consistent Matching (CCM) to mine the semantic consensus among cluster centers so that we could identify common clusters from both domains. If two cluster centers reach *consensus*, *i.e*., both centers act as the other's nearest center simultaneously, this pair will be regarded as common clusters. Secondly, we propose a metric, domain consensus score, to acquire cross-domain classification agreements between identified common clusters. Concretely, domain consensus score is defined as the proportion of samples that hold corresponding cluster label across domains. Intuitively, more samples reach consensus, the distribution shift between matched clusters is smaller. Therefore, domain consensus score could be regarded as a constraint that ensures the precise matching of CCM. Moreover, domain consensus score also offers a necessary guidance that determines the number of target clusters, and encourages the samples to be grouped into clusters of both common and private classes. Finally, for those common clusters with high domain consensus scores, we exploit a class-aware alignment technique on them to mitigate the distribution shift. As for those centers that fail to find their *consensus* counterparts, we also enhance their cluster-based consistency. To be specific, we employ a prototypical regularizer to encourage samples to approach their attached cluster centers. In this way, those samples belonging to different private categories will be encouraged to be distinguishable from each other, which also contributes to learning better representations.

Our contribution can be summarized as: 1) We tackle the UniDA problem from a new perspective, *i.e*., differentiating private samples into different clusters instead of treating them as whole. 2) We propose Domain Consensus Clustering (DCC), which mines domain consensus knowledge from two levels, *i.e*., semantic-level and sample-level, and guides the target clustering in the absence of prior knowledge. 3) Extensive experiments on four benchmarks verify the superior performance of proposed method compared with previous works.

2. Related works

Closed Set Domain Adaptation, also known as unsupervised domain adaptation (UDA), assumes two domains share identical label set. The main focus lies in how to minimize the distribution shift. Some methods minimize the discrepancy in the feature space directly [40, 39, 56, 59, 42, 61, 30, 16]. Some recent works take advantage of adversarial training to promote the alignment in the input space [11, 23, 25, 20, 10] or feature space [58, 44, 6, 18, 38, 64, 43]. Moreover, there are also some works performing adaptation via clustering in the target domain [28, 57]. However, they could not trivially generalize to the unaligned label space.

Partial Domain Adaptation (PDA) holds an assumption that private classes only lie in the source domain, which has received wide attention recently. SAN [3] employs class-wise domain discriminators to align the distributions in a fine-grained manner. IWAN [63] proposes to identify common samples with the domain similarities from the domain discriminator, and utilizes the similarities as weights for adversarial training. Recently, ETN [5] proposes a progressive weighting scheme to estimate the transferability of source samples, while BA^{3}US [36] incorporates an augmentation scheme and a complement entropy to avoid negative transfer and uncertainty propagation, respectively.

Open Set Domain Adaptation (OSDA). Different settings [54, 46, 1, 14] have been investigated for the open set domain adaptation. In this paper, we mainly focus on the setting proposed by [54], where the target domain holds private classes that are unknown to the source. OSBP [54] proposes an adversarial learning framework that enables the feature generator to learn representations to achieve common-private separation. Recent works [38, 15] follow this paradigm to draw the knowledge from the domain discriminator to identify common samples that share the semantic classes across domains. ROS [1] employs self-supervised learning technique to achieve the

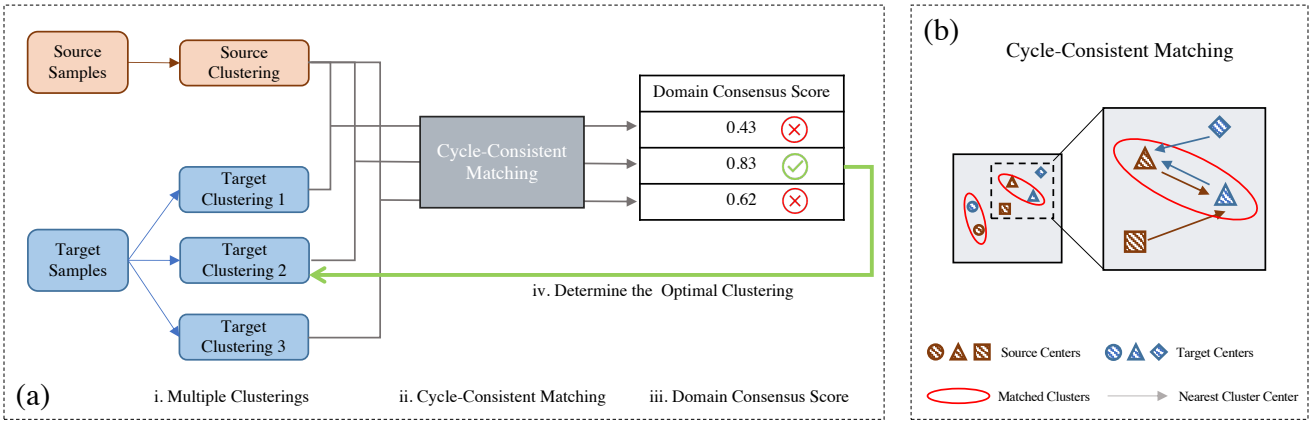


Figure 2. **(a)** Illustration of Domain Consensus Clustering (DCC). i) As the number of target classes is not given, we aim to select the optimal target clustering from multiple candidates. ii) For obtained target clusters, we leverage cycle-consistent matching (CCM) to identify clusters representing common classes from both domains. iii) Then we utilize domain consensus score to estimate the degree of agreement between matched clusters. iv) Finally, based on the domain consensus score, we could determine the optimal target clustering. **(b)** Illustration of Cycle-Consistent Matching. If two clusters from different domains act as the other's nearest neighbor, samples from the two clusters are identified as common samples that share the same semantic labels.

known/unknown separation and domain alignment.

Universal Domain Adaptation (UniDA), as a more challenging scenario, allows both domains having their own private classes. UAN [62] proposes a criterion to quantify sample-level uncertainty based on entropy and domain similarity. Then samples with lower uncertainty are encouraged for adaptation with higher weight. However, as pointed by [17], this measurement is not discriminative and robust enough. Fu *et al.* [17] designs a better criterion that combines entropy, confidence, and consistency from auxiliary classifiers to measure sample-level uncertainty. Similarly, a class-wise weighting mechanism is applied for subsequent adversarial alignment. However, they both treat private samples as one general class while ignoring the intrinsic structure of private samples.

Compared with previous works, our work differs from them mainly in two aspects. First, most previous approaches perform sample-independent evaluation to separate common samples and private samples, while we consider a cluster-based method to better exploit the intrinsic structure of target samples. Second, our approach provides a unified framework to deal with different sub-cases of universal domain adaptation, *i.e.*, OSDA, PDA, and UniDA.

3. Preliminaries

In universal domain adaptation, we are provided with annotated source samples $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n^s}$, and unlabeled target samples $\mathcal{D}^t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n^t}$. Since the label set may not be identical, we use C^s, C^t to represent label set for two domains accordingly. Then we denote $C = C^s \cap C^t$ as the common label set. We aim to train a model on $\mathcal{D}^s$ and $\mathcal{D}^t$ to classify target samples into $|C|+1$ classes, where private samples are grouped into one *unknown* class.

The model consists of two modules: (1) feature extractor f_ϕ that maps the input images into vector representation: $v = f_\phi(\boldsymbol{x})$, and (2) classifier g_ϕ that assigns each feature representation v into one of C^s classes: $p = g_\phi(v)$. For samples from two domains, we group them into clusters, respectively. The cluster assignment of source samples is based on the ground truth and the source center is the mean embedding of source samples within one specific class. For the c-th source cluster $\mathcal{D}_c^s = \{\boldsymbol{x}_i^s\}_{i=1}^{n_c^s}$, its cluster center is:

$$\mu_c^s = \frac{1}{n_c^s} \sum_{\boldsymbol{x}_i^s \in \mathcal{D}_c^s} \frac{f_\phi(\boldsymbol{x}_i^s)}{||f_\phi(\boldsymbol{x}_i^s)||}. \tag{1}$$

As for target samples, we adopt K-means to group them into K clusters and obtain corresponding centers $\{\mu_1^t, ..., \mu_K^t\}$.

4. Methodology

In this paper, we aim to utilize the domain consensus knowledge to guide the target clustering, which exploits the intrinsic structure of the target representations. Specifically, we mine the domain consensus knowledge from two levels. Firstly, the semantic-level consensus among cluster centers is utilized to identify cycle-consistent clusters as common classes (§ 4.1). Secondly, we design a novel metric named "domain consensus score" to utilize the sample-level consensus to specify the number of target clusters (§ 4.2). Finally, we discuss the cluster optimization and objectives in § 4.3. The overview of our approach is presented in Fig. 2.

4.1. Cycle-Consistent Matching

The main challenge of universal domain adaptation is how to separate common samples from private samples.

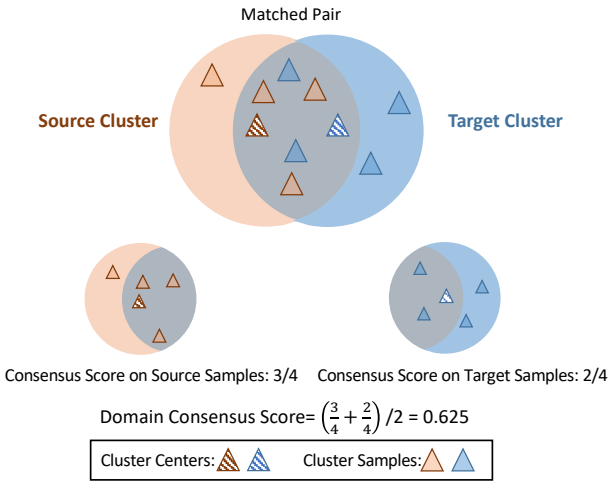


Figure 3. Illustration of Domain Consensus Score. For each sample from matched clusters, we search for its nearest cluster center in the other domain. Then domain consensus score is calculated as the proportion of samples that reach consensus, *i.e.*, the labels of their nearest cluster centers in the other domain match with those achieved by CCM.

Unlike previous works [62, 17] that perform sample-level identification on the common samples, this paper aims to mine both common classes and individual private classes simultaneously with discriminative clusters. Now a question naturally arises: *how to associate common clusters that represent the same semantic classes from both domains?* To achieve this, we propose Cycle-Consistent Matching (CCM) to link clusters from the same common classes through mining semantic-level consensus.

As illustrated in Fig. 2 (b), for each cluster center, we search for its nearest cluster center in the other domain. If two clusters reach *consensus*, *i.e.*, both act as the other's nearest center simultaneously, such a pair of clusters is recognized as common clusters. The intuition here is simple: cluster centers from the same class usually lie close enough to be associated compared to the clusters representing private classes. Further, to ensure this assumption, we utilize the sample-level consensus to promote the effectiveness of CCM, which is detailed in the next section.

4.2. Domain Consensus Score

Enabled by CCM, we could identify common samples from both domains. Nevertheless, another problem is not yet solved: *how to determine the number of target clusters without knowing the exact number of underlying target classes?* To solve this, one plausible solution is to adopt existing clustering evaluation criteria [51, 12, 2] to estimate the number of clusters. However, these techniques are designed for the single-domain scenario and cannot directly take cross-domain knowledge into consideration. Hence, we propose a metric, domain consensus score, which utilizes the sample-level consensus to determine the number of target clusters, thus forming discriminative clusters.

As shown in Fig. 3, for each sample from paired clusters, we search for its nearest cluster in the other domain, and then determine if it reaches *consensus*, *i.e.*, this sample holds corresponding cluster label across domains. Through collecting samples that reach consensus, the agreement for this pair of clusters can be evaluated.

Concretely, given a pair of matched clusters $\{v_i^s\}_{i=1}^m$ and $\{v_i^t\}_{i=1}^n$ with corresponding centers μ_c^s and μ_k^t, we aim to measure the sample-level consensus from two views, *i.e.*, the source view and the target one. To obtain consensus score on source view, for each source sample, we calculate its similarities with all target cluster centers $\{\mu_1^t, ..., \mu_K^t\}$:

$$r_{i,k}^s = Sim(v_i^s, \mu_k^t), k \in \{1, ..., K\}, \tag{2}$$

where $Sim(\cdot)$ denotes the cosine similarity, *i.e.*, $Sim(\boldsymbol{a}, \boldsymbol{b}) = \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$. Then the consensus score could be formulated as the proportion of samples that reach consensus:

$$\mathcal{S}_{(c,k)}^s = \frac{\sum_{i=1}^m \mathbb{1}\{\arg\max_k(r_{i,k}^s) = k\}}{m}, \tag{3}$$

where $\mathbb{1}\{\arg\max_k(r_{i,k}^s) = k\}$ is a indicator to judge if v_i^s holds corresponding cluster index (k) across domains.

Analogously, we could obtain the consensus score on target samples $\mathcal{S}_{(c,k)}^t$. Then we average the score of two views to obtain the consensus score of this matched pair, *i.e.*, $\mathcal{S}_{(c,k)} = \frac{\mathcal{S}_{(c,k)}^s + \mathcal{S}_{(c,k)}^t}{2}$. Finally, we calculate the mean of consensus scores of all matched pairs of clusters.

To specify the number of target clusters K, we perform multiple clusterings with different K and then we determine the optimal one according to the domain consensus score. Concretely, for different instantiations of K which are equally spaced, we compute the consensus score for each one, and the instantiation of K with the highest score is chosen for subsequent clustering.

Empirically, we find DCC tends to separate samples from one class into multiple clusters at the beginning, which is also known as over-clustering. The reason is that to achieve a higher consensus score, more accurate matching between clusters is preferred. Consequently, at the beginning, DCC prefers small clusters with "easy" samples (*i.e.* less impacted by the domain shift), which may make the number of clusters larger than the underlying number of target classes. As the adaptation proceeds, the number of clusters tends to decrease and converges to a certain number after a period of training.

4.3. Cluster Optimization and Objectives

In this section, we first introduce the clustering update strategy. Then we enumerate objectives, *i.e.*, prototypical

regularizer, contrastive domain discrepancy. Finally, we present the overall objective and the weight of each item.

Alternative Update. To avoid the accumulation of inaccurate labels, we optimize the model and update the clustering alternatively. Ideally, we expect to specify the number of clusters K with only one search, but it is impossible due to the large domain gap at the initial stage. Hence, DCC specifies the K based on domain consensus score for each update of the clustering. Empirically we find that: 1) in each round of searching, the domain consensus scores exhibit a bell curve as K increases. 2) K converges to a specific value after several initial rounds of searching, *i.e*. after early stages of training. Motivated by these observations, we adopt two stopping criteria to improve the efficiency of searching, *i.e*., stop the searching once the consensus score drops a certain number of times continuously, and fix the K once it holds a certain value for a certain number of rounds.

Prototypical Regularizer. To enhance the discriminability of target clusters, we impose a prototypical regularizer on target samples. Specifically, let $M = [\mu_1^t, \mu_2^t, ..., \mu_K^t]$ denotes the prototype bank that stores all L2-normalized target cluster centers and these prototypes is updated iteratively during training. Then the regularizer could be formulated as:

$$\mathcal{L}_{reg} = -\sum_{i=1}^{n_t}\sum_{k=1}^{K} \hat{y}_{i,k}^t \log \hat{p}_{(i,k)}, \tag{4}$$

where $\hat{y}_i^t$ is the one-hot cluster label, and

$$\hat{p}_{(i,k)} = \frac{exp({v_i}^T \mu_k^t/\tau)}{\sum_{k=1}^{K} exp({v_i}^T \mu_k^t/\tau)}. \tag{5}$$

Here v_i is L2-normalized feature vector of target samples. τ is a temperature parameter that controls the concentration of the distribution [22], and we set it to 0.1 empirically.

Contrastive Domain Discrepancy (CDD). Since the identified common samples are grouped into clusters, we leverage CDD [28, 27] to facilitate the alignment over identified common samples in a class-aware style. We impose $\mathcal{L}_{cdd}$ to minimize the intra-class discrepancies and enlarge the inter-class gap. Consequently, the enhanced discriminability, in turn, enables DCC to perform more accurate clustering. Details of CDD are provided in Appendix. A.

Overall Objective. The model is jointly optimized with three terms, *i.e*., cross-entropy loss on source samples $\mathcal{L}_{ce}$, domain alignment loss $\mathcal{L}_{cdd}$, and the regularizer $\mathcal{L}_{reg}$:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{cdd} + \gamma \mathcal{L}_{reg}, \tag{6}$$

$$\mathcal{L}_{ce} = -\sum_{i=1}^{n_s}\sum_{c=1}^{|C_s|} \hat{y}_{i,c}^s \log(\sigma(g_\phi(f_\phi(\boldsymbol{x}_i^s)))), \tag{7}$$

where σ denotes the softmax function, and $\hat{y}_i^s$ is the one-hot encoding of source label. λ is set to 0.1 for all datasets.

As mentioned, the target clustering usually converges to the optimal one after several rounds of searching, so simply applying a constant weight on $\mathcal{L}_{reg}$ may hinder the convergence as it promotes the inter-cluster separation. Therefore, we apply a ramp-up function on γ, *i.e*., $\gamma = e^{-\omega \times \frac{i}{N}}$, where i and N denote current and global iteration, and $\omega = 3.0$. Such an incremental weight allows the size of clusters to grow in the earlier stage while prevents from absorbing extra private samples after saturated.

Inference. At inference stage, we do not perform any clustering. With the prototypes $M = [\mu_1^t, \mu_2^t, ..., \mu_K^t]$, we can assign each sample a label same with the nearest prototype. In this way, common samples can be naturally separated from private ones in target domain.

5. Experiments

5.1. Setup

Besides the setting [62] where private classes exist in both domains (UniDA), we also validate our approach on other two sub-cases, *i.e*. partial domain adaptation (PDA) and open set domain adaptation (OSDA).

Dataset. We conduct experiments on four datasets. **Office-31** [52] consists of 4652 images from three domains: DSLR (**D**), Amazon (**A**), and Webcam (**W**). **Office-Home** [60] is a more challenging dataset, which consists of 15500 images from 65 categories. It is made up of 4 domains: Artistic images (**Ar**), Clip-Art images (**Cl**), Product images (**Pr**), and Real-World images (**Rw**). **VisDA** [50], is a large-scale dataset, where the source domain contains 15K synthetic images and the target domain consists of 5K images from the real world. **DomainNet** [49] is the largest domain adaptation dataset with about 0.6 million images. Like [17], we conduct experiments on three subsets from it, *i.e*., Painting (**P**), Real (**R**), and Sketch (**S**).

Following existing works [47, 54, 4, 62], we separate the label set into three parts: common classes C, source-private classes $\hat{C}_s$ and target-private classes $\hat{C}_t$. The separation of four datasets is described in Table 3. The classes are separated according to their alphabetical order.

Evaluation. For all experiments, we report the averaged results of three runs. In OSDA and UniDA, target-private classes are grouped into a single *unknown* class, and we report two metrics, *i.e*., **Acc.** and **HM**, where the former is the mean of per-class accuracy over common classes and *unknown* class, and the latter is the harmonic mean on accuracy of common samples and private ones like [17, 1]. In VisDA under OSDA, we present **OS** and **OS*** results as previous works [54, 38], where **OS** is same as **Acc.** and **OS*** only calculates the mean accuracy on common classes. In PDA, we report the mean of per-class accuracy over com-

Table 1. Results (%) on **Office-31** for UniDA (ResNet-50).

UniDA	A→W		D→W		W→D		A→D		D→A		W→A		Avg	
	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM	Acc.	HM
DANN [19]	80.65	48.82	80.94	52.73	88.07	54.87	82.67	50.18	74.82	47.69	83.54	49.33	81.78	50.60
RTN [41]	85.70	50.21	87.80	54.68	88.91	55.24	82.69	50.18	74.64	47.65	83.26	49.28	83.83	51.21
IWAN [63]	85.25	50.13	90.09	54.06	90.00	55.44	84.27	50.64	84.22	49.65	86.25	49.79	86.68	51.62
PADA [4]	85.37	49.65	79.26	52.62	90.91	55.60	81.68	50.00	55.32	42.87	82.61	49.17	79.19	49.98
ATI [47]	79.38	48.58	92.60	55.01	90.08	55.45	84.40	50.48	78.85	48.48	81.57	48.98	84.48	51.16
OSBP [54]	66.13	50.23	73.57	55.53	85.62	57.20	72.92	51.14	47.35	49.75	60.48	50.16	67.68	52.34
UAN [62]	85.62	58.61	94.77	70.62	97.99	71.42	86.50	59.68	85.45	60.11	85.12	60.34	89.24	63.46
CMU [17]	86.86	67.33	**95.72**	**79.32**	**98.01**	80.42	89.11	68.11	88.35	**71.42**	88.61	72.23	91.11	73.14
Ours	**91.66**	**78.54**	94.52	79.29	96.20	**88.58**	**93.70**	**88.50**	**90.43**	70.18	**91.97**	**75.87**	**93.08**	**80.16**

Table 2. HM (%) on **Office-Home** for UniDA (ResNet-50).

UniDA	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
DANN [19]	42.36	48.02	48.87	45.48	46.47	48.37	45.75	42.55	48.70	47.61	42.67	47.40	46.19
RTN [41]	38.41	44.65	45.70	42.64	44.06	45.48	42.56	36.79	45.50	44.56	39.79	44.53	42.89
IWAN [63]	40.54	46.96	47.78	44.97	45.06	47.59	45.81	41.43	47.55	46.29	42.49	46.54	45.25
PADA [4]	34.13	41.89	44.08	40.56	41.52	43.96	37.04	32.64	44.17	43.06	35.84	43.35	40.19
ATI [47]	39.88	45.77	46.63	44.13	44.39	46.63	44.73	41.20	46.59	45.05	41.78	45.45	44.35
OSBP [54]	39.59	45.09	46.17	45.70	45.24	46.75	45.26	40.54	45.75	45.08	41.64	46.90	44.48
UAN [62]	51.64	51.70	54.30	61.74	57.63	61.86	50.38	47.62	61.46	62.87	52.61	65.19	56.58
CMU [17]	56.02	**56.93**	**59.15**	66.95	64.27	67.82	54.72	51.09	66.39	68.24	57.89	69.73	61.60
Ours	**57.97**	54.05	58.01	**74.64**	**70.62**	**77.52**	**64.34**	**73.60**	**74.94**	**80.96**	**75.12**	**80.38**	**70.18**

Table 3. The division on label set, *i.e.*, Common Class (C) / Source-Private Class ($\hat{C}_s$) / Target Private Class ($\hat{C}_t$).

Dataset	Class Split ($\|C\|/\|\hat{C}_s\|/\|\hat{C}_t\|$)		
	PDA	OSDA	UniDA
Office-31	10 / 21 / 0	10 / 0 / 11	10 / 10 / 11
OfficeHome	25 / 40 / 0	25 / 0 / 40	10 / 5 / 50
VisDA	–	6 / 0 / 6	6 / 3 / 3
DomainNet	–	–	150 / 50 / 145

Table 4. Results (%) on **VisDA** for OSDA (VGGNet) and UniDA (ResNet-50). *: variants of OSVM using MMD and DANN.

Method	OSDA		Method	UniDA	
	OS	OS*		Acc.	HM
OSVM [26]	52.5	54.9	RTN [41]	53.92	26.02
MMD+OSVM*	54.4	56.0	IWAN [63]	58.72	27.64
DANN+OSVM*	55.5	57.8	ATI [47]	54.81	26.34
ATI-λ [47]	59.9	59.0	OSBP [54]	30.26	27.31
OSBP [54]	62.9	59.2	UAN [62]	60.83	30.47
STA [38]	66.8	63.9	USFDA [31]	63.92	–
Inheritune [32]	68.1	64.7	CMU [17]	61.42	34.64
Ours	**68.8**	**68.0**	*Ours*	**64.20**	**43.02**

mon classes.

Implementation details. Our implementation is based on PyTorch [48]. We start from ResNet-50 [21] with the backbone pretrained on ImageNet [13]. The classifier consists of two fully-connected layers, which follows the previous design [62, 17, 54, 4]. For a fair comparison, we adopt VGGNet [55] as backbone for OSDA task on VisDA.

We optimize the model using Nesterov momentum SGD with momentum of 0.9 and weight decay of 5×10^{-4}. The learning rate decays with the factor of $(1 + \alpha\frac{i}{N})^{-\beta}$, where i and N denote current iteration and global iteration, and we set $\alpha = 10$ and $\beta = 0.75$. The batch size is set to 36. The initial learning rate is set to 1×10^{-4} for Office-31 and VisDA, and 1×10^{-3} for Office-Home and DomainNet.

5.2. Comparison with previous state-of-the-arts

We compare our method with previous state-of-the-arts in three sub-cases of universal domain adaptation, *i.e.*, OSDA, PDA, and UniDA. For OSDA and PDA, we compare our method to the universal domain adaptation methods, without knowing the prior that private classes exist only in source domain (*i.e.* PDA) or only in target domain (*i.e.* OSDA). Also, we compare our method to the baselines tailed for OSDA and PDA settings, by taking the prior of each setting into consideration.

UniDA Setting. In the most challenging setting, *i.e.* UniDA, our approach achieves new state-of-the-arts. Table 1 shows the results on Office-31. The proposed method surpasses all compared methods in terms of both accuracy and HM. Especially, with respect to HM, our method outperform previous state-of-art method CMU [17] by 7%, which shows our method strikes a better balance between identifications of common and private samples. Office-Home (Table 2) is a more challenging dataset where the number of private classes is much more than common classes (55 vs. 10). Under this extreme scenario, our method demonstrates a stronger capability on the common-private separation (9% improvement in terms of HM), which benefits from the higher compactness of private sam-

Table 5. HM (%) on **Office** and **Office-Home** under the OSDA scenario (ResNet-50). The reported numbers for previous OSDA methods are cited from [1]. We use 'U' and 'O' to denote methods designed for UniDA setting and OSDA setting, respectively.

Method	Type	Office							Office-Home											
		A2W	A2D	D2W	W2D	D2A	W2A	Avg	Ar2Cl	Ar2Pr	Ar2Rw	Cl2Ar	Cl2Pr	Cl2Rw	Pr2Ar	Pr2Cl	Pr2Rw	Rw2Ar	Rw2Cl	Rw2Pr	Avg
STA_{max} [38]	O	75.9	75.0	69.8	75.2	73.2	66.1	72.5	55.8	54.0	68.3	57.4	60.4	66.8	61.9	53.2	69.5	67.1	54.5	64.5	61.1
OSBP [54]	O	82.7	82.4	**97.2**	91.1	75.1	73.7	83.7	55.1	65.2	72.9	**64.3**	64.7	**70.6**	**63.2**	53.2	73.9	66.7	54.5	72.3	64.7
ROS [1]	O	82.1	82.4	96.0	**99.7**	77.9	77.2	85.9	**60.1**	**69.3**	76.5	58.9	65.2	68.6	60.6	56.3	**74.4**	**68.8**	60.4	**75.7**	**66.2**
Ours	O	**87.1**	**85.5**	91.2	87.1	**85.5**	**84.4**	**86.8**	52.9	67.4	**80.6**	49.8	**66.6**	67.0	59.5	52.8	64.0	56.0	**76.9**	62.7	64.2
UAN [62]	U	46.8	38.9	68.8	53.0	**68.0**	54.9	55.1	0.0	0.0	0.2	0.0	0.2	0.2	0.0	0.0	0.2	0.2	0.0	0.1	0.1
Ours	U	**54.8**	**58.3**	**89.4**	**80.9**	67.2	**85.3**	**72.6**	**56.1**	**67.5**	**66.7**	**49.6**	**66.5**	**64.0**	**55.8**	**53.0**	**70.5**	**61.6**	**57.2**	**71.9**	**61.7**

Table 6. Accuracy (%) on **Office** and **Office-Home** under the PDA scenario (ResNet-50). We use 'U' and 'P' to denote methods designed for UniDA setting and PDA setting, respectively.

Method	Type	Office							Office-Home											
		A2W	A2D	D2W	W2D	D2A	W2A	Avg	Ar2Cl	Ar2Pr	Ar2Rw	Cl2Ar	Cl2Pr	Cl2Rw	Pr2Ar	Pr2Cl	Pr2Rw	Rw2Ar	Rw2Cl	Rw2Pr	Avg
IWAN [63]	P	90.5	89.2	95.6	99.3	94.3	99.4	94.7	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6
SAN [3]	P	94.3	93.9	94.2	99.3	88.7	99.4	95.0	44.4	68.7	74.6	67.5	65.0	77.8	59.8	44.7	80.1	72.1	50.2	78.7	65.3
PADA [4]	P	82.2	86.5	92.7	99.3	95.4	**100.0**	92.7	52.0	67.00	78.7	52.2	53.8	59.1	52.6	43.2	78.8	73.7	56.6	77.1	62.1
ETN [5]	P	94.5	95.0	100.0	100.0	**96.2**	94.6	96.7	**59.2**	77.0	79.5	62.9	65.7	75.0	68.3	**55.4**	84.4	75.7	57.7	84.5	70.5
RTNet [9]	P	96.2	**97.6**	100.0	100.0	92.3	95.4	96.9	63.2	80.1	80.7	66.7	69.3	77.2	71.6	53.9	**84.6**	**77.4**	57.9	85.5	72.3
Ours	P	**99.7**	96.1	**100.0**	**100.0**	95.3	96.3	**97.9**	59.0	**84.4**	**83.4**	**67.8**	**72.7**	**79.8**	**68.4**	53.2	83.7	75.8	**59.0**	**88.3**	**73.0**
UAN [62]	U	76.8	79.7	93.4	98.3	82.7	83.7	85.8	24.5	35.0	41.5	34.7	32.3	32.7	32.7	21.1	43.0	39.7	26.6	46.0	34.2
Ours	U	**97.6**	**87.3**	**100.0**	**100.0**	**96.6**	**96.3**	**96.3**	**54.2**	**47.5**	**57.5**	**83.8**	**71.6**	**86.2**	**63.7**	**65.0**	**75.2**	**85.5**	**78.2**	**82.6**	**70.9**

Table 7. HM (%) on **DomainNet** for UniDA (ResNet-50).

Method	P→R	R→P	P→S	S→P	R→S	S→R	Avg
DANN [19]	31.18	29.33	27.84	27.84	27.77	30.84	29.13
RTN [41]	32.27	30.29	28.71	28.71	28.63	31.90	30.08
IWAN [63]	35.38	33.02	31.15	31.15	31.06	34.94	32.78
PADA [4]	28.92	27.32	26.03	26.03	25.97	28.62	27.15
ATI [47]	32.59	30.57	28.96	28.96	28.89	32.21	30.36
OSBP [54]	33.60	33.03	30.55	30.53	30.61	33.65	32.00
UAN [62]	41.85	43.59	39.06	38.95	38.73	43.69	40.98
CMU [17]	50.78	**52.16**	**45.12**	44.82	**45.64**	50.97	48.25
Ours	**56.90**	50.25	43.66	**44.92**	43.31	**56.15**	**49.20**

ples. We also testify DCC on VisDA and present the results in Table 4. Notably, with higher accuracy, our method shows +9% improvement compared to CMU [17] in terms of HM, implying a higher capacity on identifying private samples. In Table 7, we present the results on a large scale dataset DomainNet. DCC yields consistent improvement, verifying its effectiveness on large-scale dataset.

OSDA and PDA setting. In Table 5, Table 6, and Table 4, we present the results under PDA and OSDA scenarios. We use 'P' and 'O' to denote the methods specifically designed for PDA and OSDA accordingly, and use 'U' to denote the UniDA methods. As shown in the tables, our approach performs favorably against previous methods in different scenarios, *i.e.* with and without using the prior knowledge. Particularly, our method without using the prior ('U') yields even better result compared to competitive methods tailed for the PDA setting. For example, on Office-Home, our method ('U') achieves 70.9% average accuracy, which outperforms PADA [4] (62.1%) and ETN [5] (70.5%), demonstrating that our method can effectively separate common samples from private ones.

5.3. Ablation Studies

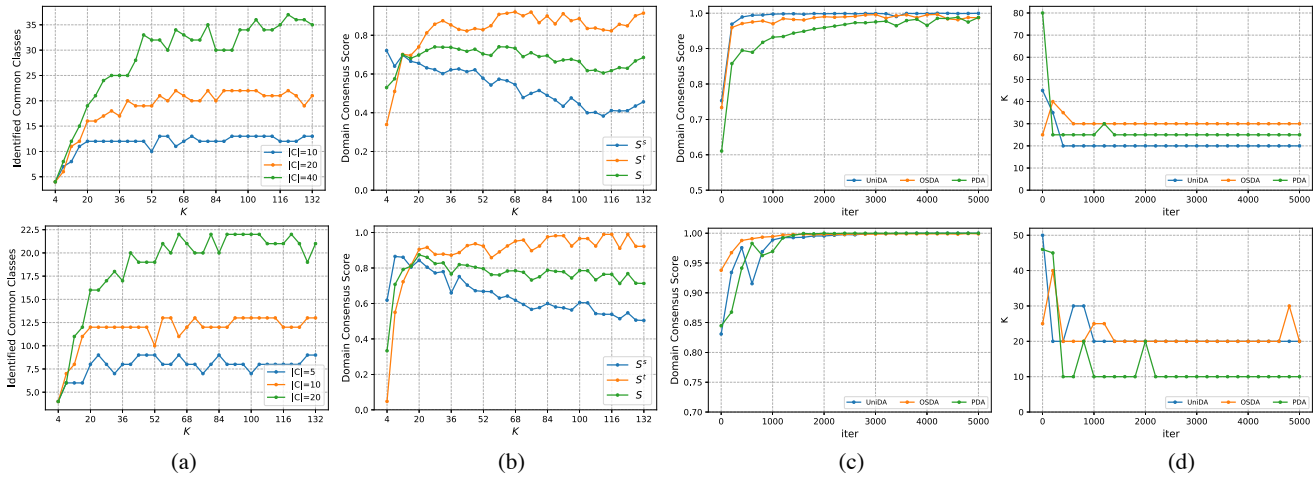
Effect of Cycle-Consistence Matching (CCM). To show how CCM can effectively identify common classes, we vary the number of common classes ($|C|$) and observe the identified common classes under different values of K. In Fig. 4 (a), as K increases, the number of matched clusters tends to converge to a value which is quite near to $|C|$.

Effect of Domain Consensus Score. To better understand domain consensus score, we conduct a series of experiments to reveal its mechanism.

First, we decompose the domain consensus score into two parts, *i.e.*, $\mathcal{S}^s$ and $\mathcal{S}^t$; $\mathcal{S}^s$ is the consensus score of source samples while the other denotes score of target samples. As shown in Fig. 4 (b), as the K increases, $\mathcal{S}^s$ and $\mathcal{S}^t$ show opposite trend, *i.e.*, $\mathcal{S}^s$ increases but $\mathcal{S}^t$ decreases. As the K increases, target samples are divided into more and smaller target clusters. Therefore, smaller target samples could better match source clusters, which causes the increase of $\mathcal{S}^t$. On the other hand, as more target clusters form, source clusters are more easily distracted by nearby target clusters, which explains the drop of $\mathcal{S}^s$.

Second, in Fig. 4 (c), we visualize the evolution of domain consensus score as training progresses. As expected, the domain consensus score saturates after the early rounds, which indicates that our method can find the optimal number of clusters quickly. Moreover, this also implies that the searching is only necessary at the early stage.

Third, we compare domain consensus score with previous general metrics to determine the number of clusters



(a) (b) (c) (d)

Figure 4. Ablation Analysis (Best viewed in color). (a) Number of identified common classes w.r.t. K under varying $|C|$. (b) Decomposed consensus score w.r.t. K. (c) The evolution of consensus score as training progresses. (d) The evolution of K as training progresses. The first row is extracted from **Ar→Rw** of Office-Home and the second row is from **A→D** of Office-31.

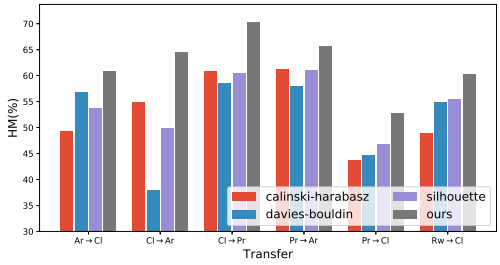


Figure 5. Performance comparison on cluster evaluation metrics.

Table 8. Effect of $\mathcal{L}_{cdd}$ and $\mathcal{L}_{reg}$.

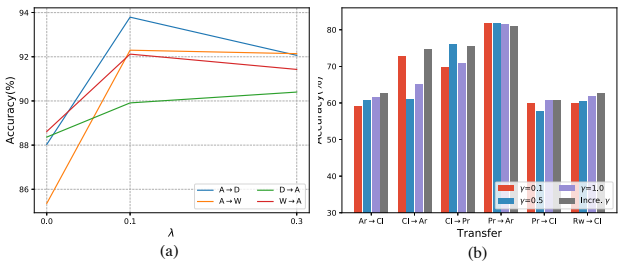
	$\mathcal{L}_{cdd}+\mathcal{L}_{reg}$	$\mathcal{L}_{cdd}$	$\mathcal{L}_{reg}$
Office-Home	**70.18**	61.48	62.85
DomainNet (P→R)	**56.90**	54.16	53.55

(*i.e.*, calinski harabasz score [2], davies bouldin score [12], and silhouette score [51]), and present the results in Fig. 5. Our metric obviously outperforms previous ones, proving the benefits of taking the distribution shift into account.

Effect of Domain Consensus Clustering. Fig. 4 (d) shows the evolution of K during training under three scenarios. In theses experiments, we do not employ the proposed stop criteria (see Section 4.3). As shown in these figures, the number of clusters converges to the optimal value after several initial searches, which is consistent with the convergence of consensus score (Fig. 4 (c)). This indicates that the searching of K is only necessary in the early stage of training, which justifies the proposed stop criteria.

Effect of $\mathcal{L}_{cdd}$ and $\mathcal{L}_{reg}$. To evaluate the contribution of $\mathcal{L}_{cdd}$ and $\mathcal{L}_{reg}$, we train the model with each component solely and present the results in Table 8, which verifies the contribution of each term.

Sensitivity to Hyper-parameters. To show the sensitivity of our method to the hyper-parameter λ, we conduct experiments on Office-31 under UniDA setting, and present



(a) (b)

Figure 6. (a) Sensitivity to λ on Office31. (b) Comparison between constant γ (*i.e.*, 0.1, 0.5, 1.0) and dynamic γ ('Incre. γ') (Office-Home). All experiments are conducted under UniDA setting.

the results in Fig. 6 (a). Within a wide range of λ (0.1-0.3), the performance only varies in a small range, showing that our method is robust to different choices of λ. Also, we compare our way of progressively increasing γ (denoted as 'Incre. γ') with using various constant values on Office-Home under UniDA setting. As shown in Fig. 6 (b), 'Incre. γ' achieves better results for most of the tasks, which verifies the effectiveness of this design.

6. Conclusion

This paper proposes Domain Consensus Clustering (DCC), which performs adaptation over unaligned label space via encouraging discriminative target clusters. To be specific, DCC exploits domain consensus knowledge from two levels, *i.e.*, semantic-level and sample-level, to identify private samples and guide the target clustering. Experiments on four benchmarks show superior performance of proposed methods, compared to previous state-of-the-arts.

Acknowledgement This work is in part supported by ARC DECRA DE190101315 and ARC DP200100938.

References

[1] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *ECCV*. Springer, 2020. 2, 5, 7

[2] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 1974. 4, 8

[3] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *CVPR*, 2018. 2, 7

[4] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *ECCV*, 2018. 1, 2, 5, 6, 7

[5] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *CVPR*, 2019. 2, 7

[6] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, 2019. 2

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 2017. 1

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 1

[9] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. Selective transfer with reinforced transfer network for partial domain adaptation. In *CVPR*, 2020. 2, 7

[10] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, June 2020. 2

[11] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016. 2

[12] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE TPAMI*, 1979. 4, 8

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 6

[14] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang. Open set domain adaptation: Theoretical bound and algorithm. *TNNLS*, 2020. 2

[15] Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or distract: Exploit the margin of open set. In *ICCV*, 2019. 2

[16] Wenkai Xu Feng Liu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *ICML*, 2020. 2

[17] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *ECCV*, 2020. 2, 3, 4, 5, 6, 7

[18] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 2

[19] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016. 6, 7

[20] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[23] 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 *ICML*, 2018. 2

[24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1

[25] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[26] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *ECCV*, 2014. 6

[27] Guoliang Kang, Lu Jiang, Yunchao Wei, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for single-and multi-source domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 5

[28] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019. 2, 5

[29] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *NeurIPS*, 2020. 1

[30] Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *ECCV*, pages 401–416, 2018. 2

[31] Jogendra Nath Kundu, Naveen Venkat, Rahul M V, and R. Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, 2020. 6

[32] Jogendra Nath Kundu, Naveen Venkat, Rahul M V, and R. Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, 2020. 6

[33] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *CVPR*, 2020. 1

[34] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *ECCV*, 2020. 1

[35] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentationg. In *NeurIPS*, 2020. 1

[36] Jian Liang, Yunbo Wang, Dapeng Hu, Ran He, and Jiashi Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *ECCV*, 2020. 2

[37] Feng Liu, Guangquan Zhang, and Jie Lu. Heterogeneous domain adaptation: An unsupervised approach. *IEEE TNNLS*, 2020. 1

[38] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, 2019. 2, 5, 6, 7

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[40] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 2

[41] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, 2016. 6, 7

[42] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *CVPR*, 2020. 2

[43] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Adversarial style mining for one-shot unsupervised domain adaptation. In *NeurIPS*, 2020. 2

[44] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *NeurIPS*, 2017. 2

[45] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010. 1

[46] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *CVPR*, 2020. 2

[47] P. Panareda Busto, A. Iqbal, and J. Gall. Open set domain adaptation for image and action recognition. *IEEE TPAMI*, 2020. 5, 6, 7

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[49] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 5

[50] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017. 5

[51] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 1987. 4, 8

[52] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 5

[53] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *NeurIPS*, 2020. 2

[54] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *ECCV*, 2018. 1, 2, 5, 6, 7

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 6

[56] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016. 2

[57] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 2020. 2

[58] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 2

[59] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, 2014. 2

[60] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation, 2017. 5

[61] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, June 2020. 2

[62] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *CVPR*, 2019. 2, 3, 4, 5, 6, 7

[63] J. Zhang, Z. Ding, W. Li, and P. Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, 2018. 2, 6, 7

[64] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision (IJCV)*, 2020. 2