Domain Consensus Clustering for Universal Domain Adaptation

Guangrui Li\textsuperscript{1}, Guoliang Kang\textsuperscript{2†}, Yi Zhu\textsuperscript{3}, Yunchao Wei\textsuperscript{1}, Yi Yang\textsuperscript{1}
\textsuperscript{1} ReLER Lab, AIIL, University of Technology Sydney
\textsuperscript{2} Carnegie Mellon University, \textsuperscript{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\textsuperscript{[55, 24]}, semantic segmentation\textsuperscript{[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)\textsuperscript{[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\textsuperscript{[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.

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)\textsuperscript{[54]} assumes the target domain possesses private classes that are unknown to the source domain. Analogously, partial domain adaptation (PDA)\textsuperscript{[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

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{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.}
\end{figure}

\textsuperscript{†} 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, 58, 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 BA3US [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
known/unknown separation and domain alignment.

Universal Domain Adaptation (UniDA), as a more challenging scenario, allows both domains having their own private classes. UAN \cite{liu2021universal} 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 \cite{fu2020universal}, this measurement is not discriminative and robust enough. Fu \textit{et al.} \cite{fu2020universal} 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, \textit{i.e.,} OSDA, PDA, and UniDA.

### 3. Preliminaries

In universal domain adaptation, we are provided with annotated source samples $D^s = \{(x^s_i, y^s_i)\}_{i=1}^n$ and unlabeled target samples $D^t = \{(x^t_i)\}_{i=1}^n$. 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 $D^s$ and $D^t$ to classify target samples into $|C| + 1$ classes, where private samples are grouped into one \textit{unknown} class.

The model consists of two modules: (1) feature extractor $f_\phi$ that maps the input images into a vector representation: $v = f_\phi(x)$, and (2) classifier $g_\psi$ that assigns each feature representation $v$ into one of $C^s$ classes: $p = g_\psi(v)$. 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 $D^s_c = \{x^s_i\}_{i=1}^{n^c}$, its cluster center is:

$$\mu^c = \frac{1}{n^c} \sum_{x^s_i \in D^s_c} f_\phi(x^s_i),$$

As for target samples, we adopt k-means to group them into $K$ clusters and obtain corresponding centers $\{\mu^1, …, \mu^K\}$.  

### 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 (\S 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 (\S 4.2). Finally, we discuss the cluster optimization and objectives in \S 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.
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 \( \mu_c^s \) and \( \mu_c^t \), 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 \( \{\mu_1, ..., \mu_K\} \):

\[
r_{i,k}^s = \text{Sim}(v_i^s, \mu_k^t), k \in \{1, ..., K\},
\]

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

\[
S_{(c,k)}^s = \frac{\sum_{i=1}^{m} I\{\arg\max_k r_{i,k}^s = k\}}{m},
\]

where \( I\{\arg\max_k r_{i,k}^s = k\} \) 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 \( S_{(c,k)}^t \). Then we average the score of two views to obtain the consensus score of this matched pair, i.e., \( S_{(c,k)} = \frac{S_{(c,k)}^s + S_{(c,k)}^t}{2} \). 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:

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

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

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

Here $v_i$ is L2-normalized feature vector of target samples. $\tau$ 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 $\mathcal{L}_{\text{cdd}}$ 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 $\mathcal{L}_{\text{reg}}$:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{cdd} + \gamma \mathcal{L}_{\text{reg}},$$  \hspace{1cm} (6)

$$\mathcal{L}_{ce} = - \sum_{i=1}^{n_s} \sum_{c=1}^{|C_t|} \hat{y}_{i,c}^s \log (\sigma (g_\phi (f_\phi(x_i^s)))$$  \hspace{1cm} (7)

where $\sigma$ denotes the softmax function, and $\hat{y}_{i,c}^s$ is the one-hot encoding of source label. $\lambda$ 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 $\mathcal{L}_{\text{reg}}$ may hinder the convergence as it promotes the inter-cluster separation. Therefore, we apply a ramp-up function on $\gamma$, i.e., $\gamma = e^{-\omega \tau}$, where $\tau$ 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 $C_s$, and target-private classes $C_t$. 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-
We optimize the model using Nesterov momentum SGD. We compare our method with previous state-of-the-arts, which benefits from the higher compactness of private sam-

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

<table>
<thead>
<tr>
<th>UniDA</th>
<th>A→W</th>
<th>D→W</th>
<th>W→D</th>
<th>A→D</th>
<th>D→A</th>
<th>W→A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>HM</td>
<td>Acc.</td>
<td>HM</td>
<td>Acc.</td>
<td>HM</td>
<td>Acc.</td>
</tr>
<tr>
<td>DANN [19]</td>
<td>80.65</td>
<td>48.82</td>
<td>80.94</td>
<td>52.73</td>
<td>88.07</td>
<td>54.87</td>
<td>82.67</td>
</tr>
<tr>
<td>RTN [41]</td>
<td>85.70</td>
<td>50.21</td>
<td>87.80</td>
<td>54.68</td>
<td>88.91</td>
<td>55.24</td>
<td>82.69</td>
</tr>
<tr>
<td>IWAN [63]</td>
<td>85.25</td>
<td>50.13</td>
<td>90.09</td>
<td>54.06</td>
<td>90.00</td>
<td>55.44</td>
<td>84.27</td>
</tr>
<tr>
<td>PADA [4]</td>
<td>85.37</td>
<td>49.65</td>
<td>97.26</td>
<td>52.62</td>
<td>90.91</td>
<td>55.60</td>
<td>81.68</td>
</tr>
<tr>
<td>ATI [47]</td>
<td>79.38</td>
<td>48.58</td>
<td>92.60</td>
<td>55.01</td>
<td>90.08</td>
<td>55.45</td>
<td>84.40</td>
</tr>
<tr>
<td>OSBP [54]</td>
<td>66.13</td>
<td>50.23</td>
<td>73.57</td>
<td>55.53</td>
<td>85.62</td>
<td>57.20</td>
<td>72.92</td>
</tr>
<tr>
<td>UAN [62]</td>
<td>85.62</td>
<td>58.61</td>
<td>94.77</td>
<td>70.62</td>
<td>97.99</td>
<td>71.42</td>
<td>86.50</td>
</tr>
<tr>
<td>CMU [17]</td>
<td>86.86</td>
<td>67.33</td>
<td>95.72</td>
<td>79.32</td>
<td>98.01</td>
<td>80.42</td>
<td>89.11</td>
</tr>
</tbody>
</table>

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 |

Table 3. The division on label set, i.e., Common Class (C) / Source-Private Class (C_p) / Target Private Class (C_t).

| Dataset | Class Split (|C|/|C_p|/|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.

<table>
<thead>
<tr>
<th>Method</th>
<th>OSDA</th>
<th>UniDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>68.8</td>
<td>68.0</td>
</tr>
<tr>
<td>OSVM [26]</td>
<td>52.5</td>
<td>54.9</td>
</tr>
<tr>
<td>MMD+OSVM*</td>
<td>54.4</td>
<td>56.0</td>
</tr>
<tr>
<td>IWAN [63]</td>
<td>58.72</td>
<td>27.64</td>
</tr>
<tr>
<td>DANN+OSVM*</td>
<td>55.5</td>
<td>57.8</td>
</tr>
<tr>
<td>ATI-λ [47]</td>
<td>59.9</td>
<td>59.0</td>
</tr>
<tr>
<td>OSBP [54]</td>
<td>30.26</td>
<td>27.31</td>
</tr>
<tr>
<td>Sta [38]</td>
<td>62.9</td>
<td>59.2</td>
</tr>
<tr>
<td>Inheriune [32]</td>
<td>66.8</td>
<td>63.9</td>
</tr>
<tr>
<td>CMU [17]</td>
<td>61.42</td>
<td>34.64</td>
</tr>
</tbody>
</table>

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 tailored 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 outperforms 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-

**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 $5 \times 10^{-4}$. The learning rate decays with the factor of $(1 + \alpha t)^{-\beta}$, where $i$ and $\beta$ 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 $1 \times 10^{-4}$ for Office-31 and VisDA, and $1 \times 10^{-3}$ for Office-Home and DomainNet.
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 D2W | Office D2A | Office D2D | Office A2W | Office A2D | Office A2A | Office D2W | Office D2A | Office D2D | Office A2W | Office A2D | Office A2A | Office-HOME D2W | Office-HOME D2A | Office-HOME D2D | Office-HOME A2W | Office-HOME A2D | Office-HOME A2A | Avg |
|------------|------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|----------------|----------------|----------------|----------------|----------------|----------------|-------|
| STRes50 [36] | 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       | U    | 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   |

<table>
<thead>
<tr>
<th>Type</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>55.1</td>
</tr>
<tr>
<td>O</td>
<td>72.6</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Office D2W</th>
<th>Office D2A</th>
<th>Office D2D</th>
<th>Office A2W</th>
<th>Office A2D</th>
<th>Office A2A</th>
<th>Office-HOME D2W</th>
<th>Office-HOME D2A</th>
<th>Office-HOME D2D</th>
<th>Office-HOME A2W</th>
<th>Office-HOME A2D</th>
<th>Office-HOME A2A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWAN [3]</td>
<td>P</td>
<td>94.3</td>
<td>94.2</td>
<td>88.7</td>
<td>99.4</td>
<td>95.0</td>
<td>94.4</td>
<td>74.6</td>
<td>67.5</td>
<td>65.0</td>
<td>77.8</td>
<td>59.8</td>
<td>54.7</td>
<td>81.3</td>
</tr>
<tr>
<td>SAN [5]</td>
<td>P</td>
<td>82.7</td>
<td>84.6</td>
<td>77.8</td>
<td>94.1</td>
<td>55.1</td>
<td>52.2</td>
<td>53.8</td>
<td>59.1</td>
<td>47.2</td>
<td>52.6</td>
<td>43.8</td>
<td>78.8</td>
<td>73.7</td>
</tr>
<tr>
<td>PADA [1]</td>
<td>P</td>
<td>94.5</td>
<td>95.0</td>
<td>100.0</td>
<td>96.2</td>
<td>94.6</td>
<td>96.7</td>
<td>62.9</td>
<td>65.7</td>
<td>75.0</td>
<td>68.3</td>
<td>55.4</td>
<td>84.4</td>
<td>75.7</td>
</tr>
<tr>
<td>ETN [5]</td>
<td>P</td>
<td>96.2</td>
<td>97.6</td>
<td>100.0</td>
<td>92.3</td>
<td>95.4</td>
<td>96.9</td>
<td>80.7</td>
<td>69.3</td>
<td>72.7</td>
<td>71.6</td>
<td>53.9</td>
<td>84.6</td>
<td>77.4</td>
</tr>
<tr>
<td>RTNet [5]</td>
<td>P</td>
<td>90.7</td>
<td>96.1</td>
<td>100.0</td>
<td>95.3</td>
<td>96.3</td>
<td>97.9</td>
<td>65.4</td>
<td>67.8</td>
<td>72.7</td>
<td>75.8</td>
<td>58.4</td>
<td>83.7</td>
<td>77.5</td>
</tr>
<tr>
<td>Ours</td>
<td>U</td>
<td>78.8</td>
<td>79.7</td>
<td>93.4</td>
<td>83.7</td>
<td>83.7</td>
<td>85.8</td>
<td>45.5</td>
<td>34.7</td>
<td>32.3</td>
<td>32.1</td>
<td>57.4</td>
<td>17.4</td>
<td>17.4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>R→P</th>
<th>R→S</th>
<th>P→S</th>
<th>P→R</th>
<th>S→P</th>
<th>S→R</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>DANN [19]</td>
<td>31.8</td>
<td>28.4</td>
<td>27.8</td>
<td>28.7</td>
<td>27.7</td>
<td>30.8</td>
<td>29.13</td>
</tr>
<tr>
<td>RTN [41]</td>
<td>32.7</td>
<td>30.2</td>
<td>28.7</td>
<td>28.7</td>
<td>28.6</td>
<td>31.9</td>
<td>30.08</td>
</tr>
<tr>
<td>IWAN [63]</td>
<td>35.8</td>
<td>33.0</td>
<td>31.5</td>
<td>31.5</td>
<td>31.0</td>
<td>34.9</td>
<td>32.78</td>
</tr>
<tr>
<td>PADA [5]</td>
<td>28.9</td>
<td>27.3</td>
<td>26.3</td>
<td>25.9</td>
<td>25.7</td>
<td>28.2</td>
<td>27.15</td>
</tr>
<tr>
<td>ATI [47]</td>
<td>32.5</td>
<td>30.5</td>
<td>28.6</td>
<td>28.6</td>
<td>28.8</td>
<td>32.2</td>
<td>30.36</td>
</tr>
<tr>
<td>OSBP [54]</td>
<td>33.6</td>
<td>33.0</td>
<td>30.5</td>
<td>30.5</td>
<td>30.6</td>
<td>33.6</td>
<td>32.00</td>
</tr>
<tr>
<td>UAN [62]</td>
<td>41.8</td>
<td>43.5</td>
<td>39.6</td>
<td>38.9</td>
<td>38.7</td>
<td>43.6</td>
<td>40.98</td>
</tr>
<tr>
<td>CMU [17]</td>
<td>50.7</td>
<td>52.1</td>
<td>45.2</td>
<td>45.2</td>
<td>45.4</td>
<td>50.9</td>
<td>48.25</td>
</tr>
</tbody>
</table>

| Ours       | 56.90 | 50.25 | 43.66 | 44.92 | 45.31 | 56.15 | 49.20  |

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.

Table 8. Ablation Studies

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., S^t and S^t; S^t 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, S^t and S^t show opposite trend, i.e., S^t increases but 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 explains the drop of S^t.

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.
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 $\text{Ar}\rightarrow\text{Rw}$ of Office-Home and the second row is from $\text{A}\rightarrow\text{D}$ of Office-31.

Figure 5. Performance comparison on cluster evaluation metrics.

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

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{L}<em>{\text{cdd}}$ + $\mathcal{L}</em>{\text{reg}}$</th>
<th>$\mathcal{L}_{\text{cdd}}$</th>
<th>$\mathcal{L}_{\text{reg}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office-Home</td>
<td>70.18</td>
<td>61.48</td>
<td>62.85</td>
</tr>
<tr>
<td>DomainNet (P→R)</td>
<td>56.90</td>
<td>54.16</td>
<td>53.55</td>
</tr>
</tbody>
</table>

(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 these 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 $\mathcal{L}_{\text{cdd}}$ and $\mathcal{L}_{\text{reg}}$. To evaluate the contribution of $\mathcal{L}_{\text{cdd}}$ and $\mathcal{L}_{\text{reg}}$, 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 $\lambda$, we conduct experiments on Office-31 under UniDA setting, and present the results in Fig. 6 (a). Within a wide range of $\lambda$ (0.1-0.3), the performance only varies in a small range, showing that our method is robust to different choices of $\lambda$. Also, we compare our way of progressively increasing $\gamma$ (denoted as ‘Incre. $\gamma$’) with using various constant values on Office-Home under UniDA setting. As shown in Fig. 6 (b), ‘Incre. $\gamma$’ 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


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


