Collaborative Semantic Consistency Alignment for Blended-Target Domain Adaptation

Yuwu Lu[1, 2], **Xue Hu**[1], **Waikeung Wong**[2*], **Haoyu Huang**[1]

[1]South China Normal University, Guangzhou, China
[2]Hong Kong Polytechnic University, Hong Kong, China
{luyuwu2008, hx1430940232, hyhuang99}@163.com, calvin.wong@polyu.edu.hk

Abstract

Blended-target domain adaptation (BTDA) leverages learned source knowledge to adapt the model to a blended-target domain that is composed of multiple unlabeled sub-target domains with distinct statistical characteristics. The existing BTDA methods usually overlook semantic correlation information across multiple domains and domain shifts among sub-target domains, resulting in sub-optimal adaptation performance. To fully harness semantic knowledge and alleviate domain shifts in hybrid data distribution, we propose a collaborative semantic consistency alignment (CSCA) method for BTDA. Specifically, we achieve distribution alignment by minimizing the sliced Wasserstein distance between the source and target feature distributions. To alleviate complex domain shifts among all sub-target domains in the hybrid feature space, we design graph networks to propagate and share semantic knowledge across domains, which reduce semantic discrepancies among multiple domains. Additionally, we propose a double consistency regularization method to reduce the susceptibility of the model to domain-specific information, further facilitating semantic alignment and alleviating domain shifts. Extensive experiments on several datasets show that CSCA achieves promising classification performance.

Code — https://github.com/xuehu365/CSCA

Introduction

Unsupervised domain adaptation (UDA) (Long et al. 2015) aims to improve the generalization on the unlabeled target domain by transferring knowledge from the labeled source domain. Most UDA methods are usually confined to adapting from a single source to a single target (STDA) (Bai et al. 2024; Xie et al. 2022; Han, Sun, and Yin 2022). However, in real-world applications, the target data may originate from various environments with significant distribution disparities. For example, in remote sensing image classification, the images in a dataset are collected under diverse conditions, including geographical positions, atmospheric interference, and different satellite sensor configurations. To adapt to real-life situations, a more challenging DA task known as blended-target domain adaptation (BTDA) (Chen et al. 2019; Xu, Wang, and Ling 2023; Peng et al. 2019) has received significant attention.

As a special case of multi-target domain adaptation (MTDA) (Nguyen-Meidine et al. 2021; Zhou et al. 2023; Kiran et al. 2022), BTDA adapts from a source domain to a blended-target domain, where all target domains are mixed together without any domain labels. Thus, it is difficult to distinguish which target domain these samples come from. In other MTDA methods (Isobe et al. 2021; Roy et al. 2021; Tian et al. 2022), target domains are independent with distinct identity IDs known as domain labels. Compared with MTDA methods with domain labels, BTDA has the following advantages: (1) mixed targets are beneficial for simultaneously capturing semantic information among all domains without the limitations of domain isolation. (2) MTDA methods are susceptible to experiencing catastrophic forgetting of previously-trained target domains, which rely on the sequence of target adaptations (Ngo et al. 2023). However, there is no such problem in BTDA methods because all target samples from distinct domains are mixed together and trained simultaneously.

Although BTDA has some merits, there are relatively few studies on it at present because the BTDA scenario is complicated and challenging for the following reasons. Primarily, domain shifts exist not only between the source and target domains but also among all implicit sub-target domains. In addition, the hybrid data distribution leads to a serious category mismatching phenomenon where the clusters of different classes from different domains may overlap in the hybrid feature space (Xu, Wang, and Ling 2023). To address these challenges, Chen et al. (2019) employed an unsupervised meta-learner to partition the combined target into multiple clusters, termed as meta-sub-targets. The model automatically devises the multi-target adaptation losses for each meta-sub-target to mitigate domain shifts. Although Chen et al. solved the category mismatching problem, they failed to leverage the advantage (1) of the blended-target domain adaptation, but instead employed a meta-learner to separate the mixed target domain into multiple independent sub-target domains, making it difficult to explore correlation information among target domains. Xu et al. (2023) used a category domain discriminator, guided by uncertainty estimation, to align categorical distributions. And they augmented the source data with target styles to facilitate semantic trans-

*Corresponding author.

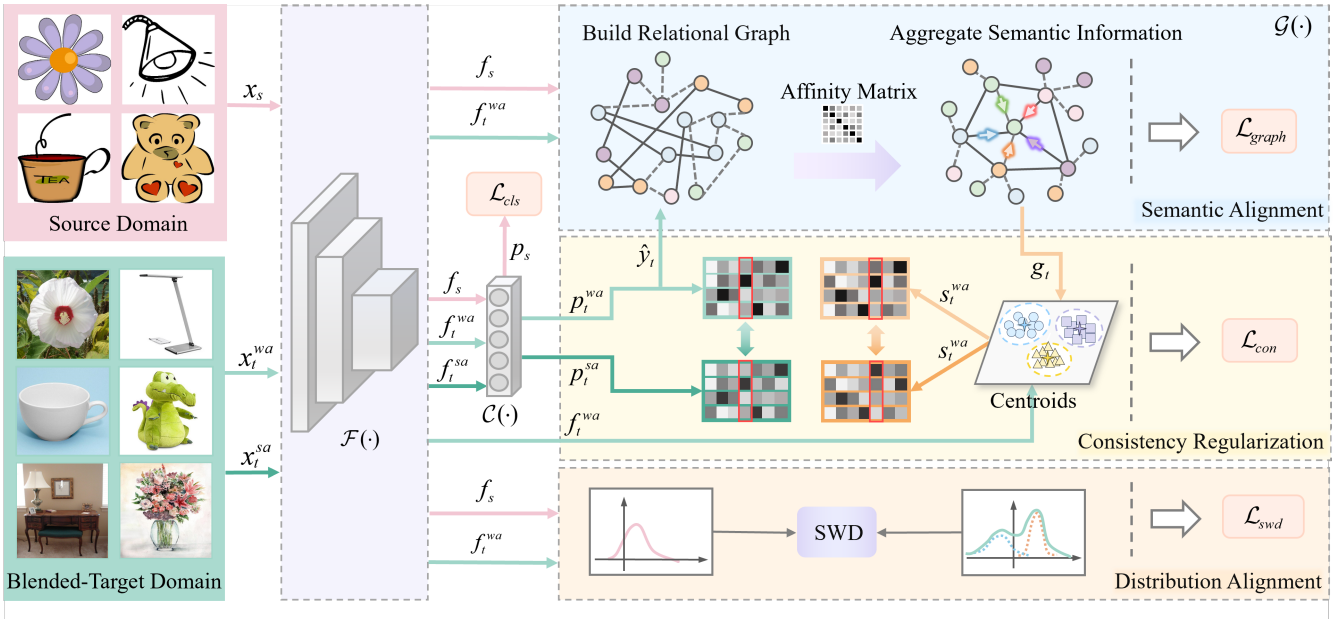


Figure 1: Illustration of the proposed collaborative semantic consistency alignment approach. $\mathcal{F}(\cdot)$ is the feature extractor, $\mathcal{C}(\cdot)$ is the MLP classifier, and $\mathcal{G}(\cdot)$ is the graph network module. g_t is a batch-size of prediction probabilities from $\mathcal{G}(\cdot)$. $\hat{y}_t$ is a batch-size of pseudo-labels from $\mathcal{C}(\cdot)$. For the inter-domain distribution alignment, we minimize the sliced Wasserstein distance between the source and target feature distributions. For the cross-domain semantic alignment, we design graph networks to conduct semantic knowledge exchange and information sharing among all domains. In order to appear more concise, we have hidden some edges in the fully connected graph. For the intra-domain double consistency regularization, we keep the consistency of two views from both prediction and feature aspects to reduce the sensitivity of the model to untransferable information.

fer from the target to the source, effectively mitigating semantic discrepancies. Although Xu et al. considered the semantic information between source and target, they ignored the semantic correlations among the implicit sub-target domains, which are also essential for achieving fine-grained domain adaptation.

To make up for the aforementioned deficiencies and address the challenges of BTDA, we propose a collaborative semantic consistency alignment (CSCA) method for BTDA. Specifically, we first align global distributions between the source and blended-target domains by minimizing the sliced Wasserstein distance. Meanwhile, to alleviate the domain shifts among all implicit sub-target domains, we design graph networks to conduct knowledge exchange and information sharing not only between the source and target domains but also among implicit sub-target domains. In this process, the graph networks can generate more discriminative and robust features that aggregate semantic properties from similar features across domains. This semantic characteristic aggregation method can solve category mismatching problems because the similar samples from different domains tend to cluster more tightly and the dissimilar samples tend to move further apart. However, the untransferable semantic information, such as domain-specific information, will be aggregated into new learned representations, which ultimately harm the generalization ability of models (Wiles et al. 2021). To reduce the interference of domain-specific information and enhance the robustness of the model in complex mixed domains, we propose a novel double consistency regularization method, ensuring consistency in category assignments (from the prediction aspect) and centroid assignments (from the feature aspect). As shown in Fig. 1, we integrate the distribution alignment and the semantic matching into a unified BTDA framework. The contributions in the paper can be outlined as follows:

- We propose a novel approach for BTDA that simultaneously considers domain distribution alignment and semantic alignment for comprehensive adaptation. These two factors complement each other in a collaborative training manner.
- We design graph networks to disseminate and exchange semantic information across all domains, which effectively reduces semantic discrepancies among different domains.
- We introduce a double consistency regularization method to reduce the model's susceptibility to domain-specific information, which is beneficial for learning domain-invariant features and improving the generalization capability of the model.

Related Work

Multi-Target Domain Adaptation

Multi-target domain adaptation (MTDA) is a transfer learning scenario that migrates knowledge from a labeled source domain to multiple unlabeled target domains with different

distributions. Current MTDA methods can generally be divided into two types: MTDA methods with domain labels (Isobe et al. 2021; Zhou et al. 2023; Kiran et al. 2022) and blended-target domain adaptation (BTDA) methods (Xu, Wang, and Ling 2023; Chen et al. 2019; Peng et al. 2019). BTDA is an emerging domain-agnostic transfer scenario in which all sub-target domains are blended as a single target domain. At present, the research on BTDA is highly appealing due to the diversity of sources for target domain data. Xu et al. (2023) designed a categorical domain discriminator with the guidance of uncertainty to align the category distributions in the hybrid feature space. Chen et al. (2019) utilized a meta-learner to automatically devise adversarial meta-sub-target DA losses, which can dynamically train the model to learn domain-invariant features. However, they fail to explore the semantic correlations among various target domains, which is important in alleviating multi-domain shifts. Therefore, we design graph networks to achieve semantic information exchange across all domains and learn more robust and discriminative features. We also propose a double consistency regularization method to reduce the interference of domain-specific information on the model during information exchange.

Graph Neural Networks

Graph neural networks (GNNs) (Wu et al. 2021) can effectively capture the intricate relationships and facilitate the exchange of messages among the nodes in a graph. Due to their proficiency in feature aggregation and information propagation, GNNs has been widely employed in domain adaptation tasks (Zhang et al. 2024; Yang et al. 2020; Yuan et al. 2022). For instance, Yuan et al. (2022) proposed a self-supervised learning strategy that GNNs serves as a bridge between the pretext task (domain classification) and the target task (category classification) to achieve transferable information sharing. However, this strategy is not suitable for the BTDA setup due to the lack of domain labels for domain classification. Yang et al. (2020) proposed a heterogeneous graph attention network (HGAN) to disseminate semantic information between the source and target. However, in this method, HGAN only conducts semantic propagation between the source and a single sub-target domain, which neglects the domain shifts among the sub-target domains. Differently, we leverage GNNs to achieve semantic information propagation not only between the source and target domains but also among all sub-target domains, which can effectively reduce the semantic discrepancies among multiple domains.

Consistency Regularization

Consistency regularization (Kurakin, Goodfellow, and Bengio 2017) is a self-supervised learning (Chen et al. 2020) technique used to improve the generalization performance of models without any annotations. In domain adaptation, consistency regularization can make the models less sensitive to domain-specific features, enabling them to maintain robustness against task-irrelevant information (Jing et al. 2023). Most consistency regularization methods (Xie et al. 2020; Sohn et al. 2020; Yan et al. 2022) are compelled to produce consistent representations or predictions for two different augmented versions of an image. However, directly imposing constraints on predictions or features can be overly strict, potentially complicating the training process (Jing et al. 2023). In our work, we propose a novel double consistency regularization coupled with clustering centroids to reduce the sensitivity of the model to domain-specific information. We simultaneously consider the perspectives of prediction probability and centroid similarity without directly imposing constraints on them.

Method

Preliminaries

Notation. In the BTDA scenario, we are given a labeled source domain $\mathcal{S} = \{(x_{si}, y_{si})\}_{i=1}^{n_s}$ with n_s labeled sample pairs and a combined target domain $\mathcal{T} = \{\mathcal{T}_k\}_{k=1}^{K} = \{x_{tj}\}_{j=1}^{n_t}$ with K unlabeled sub-target domains, where n_t is the number of all target samples. x_{si} is a source image sample, and y_{si} is the corresponding ground truth label. The source domain and all sub-target domains have a common label space, but they have different statistical properties. Not only are there domain shifts between the source domain and all sub-target domains, but also the data distributions differ among all sub-target domains themselves. Our BTDA model consists of a feature extractor $\mathcal{F}(\cdot)$, a MLP classifier $\mathcal{C}(\cdot)$, and graph networks $\mathcal{G}(\cdot)$. $\mathcal{F}(\cdot) : \mathbb{R}^{3\times w\times h} \rightarrow \mathbb{R}^d$, parameterized by $\theta_{\mathcal{F}}$, is used to extract source and target features, i.e., $f_s = \mathcal{F}(x_s)$ and $f_t = \mathcal{F}(x_t)$. $\mathcal{C}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_c}$, parameterized by $\theta_{\mathcal{C}}$, is used to output class logits p'_s and p'_t that pass through the softmax function to obtain probabilities p_s and p_t. n_c is the number of categories in the dataset. $\mathcal{G}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{n_c}$, parameterized by $\theta_{\mathcal{G}}$, outputs class logits g'_s and g'_t that pass through the softmax function to obtain probabilities g_s and g_t.

The Design of Graph Networks. $\mathcal{G}(\cdot)$ comprises an edge relationship network $h_e(\cdot)$ and a node classification network $h_v(\cdot)$. For an undirected fully connected graph $\mathcal{I} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ constructed by a batch-size of samples, $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and $\mathcal{A}$ is an affinity matrix. $v_i \in \mathcal{V}$ is a node in $\mathcal{I}$, which is initialized by the extracted feature from $\mathcal{F}(\cdot)$. $e_{i,j} \in \mathcal{E}$ denotes the edge between nodes v_i and v_j. The matrix $\mathcal{A}$ is formed by many similarity scores, where each score $a_{i,j} \in \mathcal{A}$ reflects the semantic similarity between feature nodes v_i and v_j. We obtain the similarity score $\hat{a}_{i,j}^{(l)}$ for the pair $(v_i, v_j) \in \mathcal{E}$ at the l-th layer using the following equation:

$$\hat{a}_{i,j}^{(l)} = h_e^{(l)}(v_i^{(l-1)}, v_j^{(l-1)}), \tag{1}$$

where $h_e^{(l)}$ is the l-th layer of the edge relationship network $h_e(\cdot)$ that calculates semantic similarity between feature node pairs. $v_i^{(l-1)}$ is the node feature at the $(l-1)$-th GCNs layer for the data x_i. Then, we add self-loops for each feature node and normalize the learned similarity scores to form the affinity matrix $\mathcal{A}^{(l)}$:

$$\mathcal{A}^{(l)} = D^{-\frac{1}{2}}(\hat{\mathcal{A}}^{(l)} + I)D^{-\frac{1}{2}}, \tag{2}$$

where D is the degree matrix, I is the identity matrix, and $\hat{\mathcal{A}}$ is the unnormalized affinity matrix. When we obtain $\mathcal{A}^{(l)}$, we can conduct information exchange among node embeddings and update the node features with the following feature aggregation formula:

$$v_i^{(l)} = h_v^{(l)}([v_i^{(l-1)}, \sum_{j \in \mathcal{B}} a_{i,j}^{(l)} \cdot v_j^{(l-1)}]), \tag{3}$$

where $h_v^{(l)}$ is the l-th layer of feature aggregation network $h_v(\cdot)$, and the last layer of $h_v(\cdot)$ is a classification layer with n_c outputs. $[\cdot,\cdot]$ is the concentration operator. $\mathcal{B}$ is the set of sample indexes in a mini-batch. v_i is the current node feature that needs to aggregate semantic information from others, and v_j is other node feature in the same batch-size.

Supervised Learning for Labeled Data. To effectively learn source domain knowledge, we optimize the classification cross-entropy losses for the labeled source data:

$$\mathcal{L}_{cls} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \tilde{y}_{si} \log(\mathcal{C}(\mathcal{F}(x_{si}))), \tag{4}$$

$$\mathcal{L}_{v} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \tilde{y}_{si} \log(h_v(\mathcal{F}(x_{si}))), \tag{5}$$

where $\tilde{y}_{si}$ is a source one-hot label. The MLP classification loss $\mathcal{L}_{cls}$ is used to optimize $\mathcal{C}(\cdot)$, and the node classification loss $\mathcal{L}_v$ is used to optimize $h_v(\cdot)$.

Inter-domain Distribution Alignment

The Wasserstein distance can effectively capture the geometrical properties inherent in distributions and exhibit greater stability. However, when dealing with high-dimensional distributions, the computational complexity will become extremely high, leading to low efficiency or even infeasibility of computation. Investigated in (Lee et al. 2019; Xie et al. 2022), we use a simpler and more efficient variational version: the sliced Wasserstein distance (SWD) as our domain discrepancy measurement, which simplifies the complex high-dimensional distribution into several one-dimensional distributions. The SWD of feature distributions between the source and the target is described as follows:

$$\mathcal{L}_{swd}(f_s, f_t) = \sum_{m=1}^{M} \sum_{i=1}^{N} \psi(\xi(\langle f_s, \alpha_m \rangle)_i, \xi(\langle f_t, \alpha_m \rangle)_i), \tag{6}$$

where $\psi(\cdot,\cdot)$ is the adaptation cost function, $\xi(\cdot)$ is the sorting function, $\langle \cdot,\cdot \rangle$ is the inner-product, and N is the number of samples in a batch-size. We set the number of projections $M = 256$, as done in (Lee et al. 2019). α_m is a projection vector randomly sampled from the unit sphere $\mathbb{S}^{d-1}$, i.e., $\alpha_m \in \mathbb{S}^{d-1}$. d is the dimension of features f_s and f_t. By minimizing the above domain alignment loss (in Eq. (6)), we can effectively reduce domain shifts and capture the geometrically meaningful discrepancies between the source and target features.

Cross-domain Semantic Alignment

Domain alignment solely reduces the global distribution discrepancy but overlooks the semantic characteristics contained in the data, which may compromise the class discriminative ability (Ma, Zhang, and Xu 2019). Additionally, the domain shifts among the implicit sub-target domains are not resolved either. To address these concerns, we design graph networks to conduct semantic information exchange and knowledge sharing across multiple domains in the hybrid feature space.

As mentioned earlier, the edge relationship network $h_e(\cdot)$ outputs an unnormalized affinity matrix $\hat{\mathcal{A}}$ using Eq. (1), where each term in the matrix denotes the similarity score between a pair of node features. So how can we teach the network $h_e(\cdot)$ effectively to learn the similarity between the node features? Firstly, we need to construct an association matrix $\mathcal{A}^*$ to guide $h_e(\cdot)$ learning, and each element $a_{i,j}^*$ in $\mathcal{A}^*$ represents whether sample x_i and x_j belong to the same class, which is described as follows:

$$a_{i,j}^* = \begin{cases} 1, & \texttt{if } y_i = y_j \\ 0, & \texttt{otherwise} \end{cases}. \tag{7}$$

If the class labels of two samples are identical, then the corresponding association element $a_{i,j}^*$ is assigned the value "1", indicating that an edge is built between x_i and x_j. For the source data, class labels are already available. For unannotated target data, we use pseudo-labels provided by the MLP classifier $\mathcal{C}(\cdot)$ to establish the matrix $\mathcal{A}^*$. $\mathcal{C}(\cdot)$ does not aggregate features before classification, so it is not affected by unsimilar sample noise in the early stage.

To instruct $h_e(\cdot)$ to learn the sample similarities in a batch-size data, we align its outputs, the unnormalized affinity matrix $\hat{\mathcal{A}}$, as closely as possible with $\mathcal{A}^*$. We optimize $h_e(\cdot)$ by using a binary cross-entropy loss:

$$\mathcal{L}_e = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} [a_{i,j}^* \log(\hat{a}_{i,j}) + (1 - a_{i,j}^*) \log(1 - \hat{a}_{i,j})], \tag{8}$$

where n is the total number of a batch-size source and target samples. To reduce the impact of unreliable pseudo-labels, the elements a^* about low-confidence pseudo-labels in $\mathcal{A}^*$ (the maximum likelihood of prediction is less than threshold τ_1) are not optimized during training.

After edge relationship network $h_e(\cdot)$ outputs $\hat{\mathcal{A}}$, we normalize $\hat{\mathcal{A}}$ using Eq. (2) to obtain affinity matrix $\mathcal{A}$. Then, the node classification network $h_v(\cdot)$ will conduct information exchange among the node embeddings to learn semantically rich features based on $\mathcal{A}$ using Eq. (3). During the feature updating process, highly similar features contribute more to the formation of the final features. Through the knowledge aggregation of multi-layer GCNs, we can obtain more discriminative features. At the last layer of $h_v(\cdot)$, the network will classify the samples by outputting n_c-dimensional vectors. We optimize $h_v(\cdot)$ by using Eq. (5). Finally, the graph networks optimization loss is denoted as follows:

$$\mathcal{L}_{graph} = \lambda_e \mathcal{L}_e + \lambda_v \mathcal{L}_v, \tag{9}$$

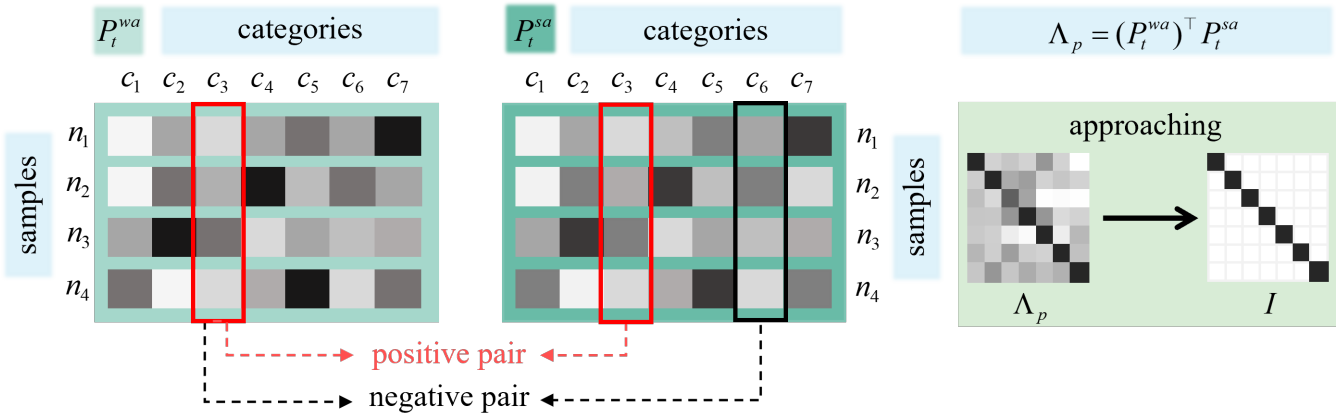


Figure 2: Illustration of the consistency of category assignments. Category assignments of the same class from two views can form positive pairs, while those from different classes can form negative pairs.

where λ_e and λ_v are balance parameters. In this semantic-oriented alignment, we can capture the intrinsic semantic information behind the data to mitigate domain shifts among the implicit sub-target domains, which can relieve the phenomenon of hybrid feature distribution.

Intra-domain Double Consistency Regularization

To decrease the interference of domain-specific information and promote intra-domain representation learning, we conduct consistency regularization on target data from double aspects. Formally, for each batch-size of target samples x_t, we create two views: a weakly augmented view x_t^{wa} and a strongly augmented view x_t^{sa}. Then, two views pass through $\mathcal{F}(\cdot)$ to get their feature representations f_t^{wa} and f_t^{sa}. The features are finally fed into $\mathcal{C}(\cdot)$ with the softmax function $\sigma(\cdot)$ to obtain the probability predictions P_t^{wa} and P_t^{sa}. This procedure is described as follows:

$$P_t^{view} = \sigma(\mathcal{C}(f_t^{view})) = \sigma(\mathcal{C}(\mathcal{F}(x_t^{view}))), \tag{10}$$

where $view$ is wa (weak augment) or sa (strong augment). We regard each column of P_t^{view} as the category assignment (content boxed in red in Fig. 2), i.e., a probability vector of a batch-size of samples being assigned to a certain category.

Our double consistency regularization method both considers the consistency between two views from prediction and feature aspects. For the prediction aspect, we increase consistency between the category assignments of the same class from two views (positive pairs) and decrease the similarity between the category assignments of the different classes from two views (negative pairs). The prediction relevance matrix $\Lambda_p \in \mathbb{R}^{n_c \times n_c}$ measures category assignment similarities, which is calculated using the following formula:

$$\Lambda_p = (P_t^{wa})^\top P_t^{sa}, \tag{11}$$

where each element $\Lambda_p^{i,j}$ in the asymmetry matrix Λ_p evaluates the similarity between the i-th column of P_t^{wa} and the j-th column of P_t^{sa}. The objective is to increase the similarity of positive pairs (diagonal value in Λ_p) and decrease the similarity of negative pairs (off-diagonal values), so we can get the optimize function as:

$$\mathcal{L} = \frac{1}{2n_c}(\|\phi(\Lambda_p) - I\|_1 + \left\|\phi(\Lambda_p^\top) - I\right\|_1), \tag{12}$$

where $\phi(\cdot)$ is a normalization function ensuring that the sum of values in each row equals 1. I is an identity matrix. $\|\cdot\|_1$ is the operation of summing the absolute values of the matrix.

From the feature aspect, we first compute similarity between feature f_{ti} and centroid c_r with the equation below:

$$s_{i,r} = \frac{\exp(sim(f_{ti}, c_r)/\tau_2)}{\sum_{j=1}^{n_t} \exp(sim(f_{tj}, c_r)/\tau_2)}, \tag{13}$$

where $sim(\cdot,\cdot)$ denotes the cosine similarity function and τ_2 is a scaling temperature. After calculating similarities between a batch-size of features and all centroids, we can obtain similarity matrices S_t^{wa} and S_t^{sa} from two views. Similar to Λ_p, the centroid relevance matrix $\Lambda_f \in \mathbb{R}^{n_c \times n_c}$ measures centroid assignment similarities between two views:

$$\Lambda_f = (S_t^{wa})^\top S_t^{sa}. \tag{14}$$

We aim to increase consistency between the centroid assignments of the same class from two views, and reduce the similarity between the centroid assignments of the different classes from two views. From the above, our double consistency loss can be defined as:

$$\begin{aligned}\mathcal{L}_{con} = \frac{1}{2n_c}[&\lambda_p(\|\phi(\Lambda_p) - I\|_1 + \left\|\phi(\Lambda_p^\top) - I\right\|_1) \\ &+ \lambda_f(\|\phi(\Lambda_f) - I\|_1 + \left\|\phi(\Lambda_f^\top) - I\right\|_1)],\end{aligned} \tag{15}$$

where λ_p and λ_f are weighting parameters. It is noteworthy that we use a method like weighted k-means clustering in (Liang, Hu, and Feng 2020) to calculate the centroids $\{c_r\}_{r=1}^{n_c}$ with the robust predictions from $\mathcal{G}(\cdot)$.

Training Procedure

Our model undergoes three training stages. The first stage is the pre-training phase, in which we train the model using labeled source data. The second stage is the domain adaptation

Methods	Office-31				Office-Home				
	A	D	W	Avg.	Ar	Cl	Pr	Rw	Avg.
Source only	68.6	70.0	66.5	68.4	47.6	42.6	44.2	51.3	46.4
DAN	79.5	80.3	81.2	80.4	55.6	56.6	48.5	56.7	54.4
DANN	80.8	82.5	83.2	82.2	58.4	58.1	52.9	62.1	57.9
CDAN	93.6	80.5	81.3	85.1	59.5	61.0	54.7	62.9	59.5
JAN	84.2	74.4	72.0	76.9	58.3	60.5	52.2	57.5	57.1
AMEAN	90.1	77.0	73.4	80.2	64.3	65.5	59.5	66.7	64.0
CGCT	93.9	85.1	85.6	88.2	67.4	68.1	61.6	68.7	66.5
MT-MTDA	87.9	83.7	84.0	85.2	64.6	66.4	59.2	67.1	64.3
DCGCT	93.4	86.0	87.1	88.8	70.5	70.5	66.0	71.2	69.8
MTDA-PDT	92.9	85.9	87.3	88.7	71.1	73.0	67.5	72.1	71.9
MCDA	92.4	87.7	88.8	89.6	71.7	72.8	68.0	71.7	71.1
CSCA(ours)	**95.8**	**88.1**	**89.2**	**91.0**	**73.5**	**75.0**	**69.7**	**73.6**	**73.0**

Table 1: Accuracy (%) of BTDA on the Office-31 and the Office-Home datasets. Each domain in the table represents the source, and the remaining domains are mixed as the target. The accuracy is the mean of accuracies across all sub-target domains in the blended target. The best results are highlighted in bold.

phase, with the overall optimization objective function:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda_{swd}\mathcal{L}_{swd} + \mathcal{L}_{graph} + \mathcal{L}_{con}, \quad (16)$$

where λ_{swd} is a trade-off parameter. In this stage, we adopt an episodic training scheme to gradually update the target domain by assigning reliable pseudo-labels to the corresponding target samples in each episode. Because GCNs can provide more reliable and robust pseudo-labels, we use high-confidence pseudo-labels (the maximum likelihood of prediction is greater than threshold τ_1) predicted by $\mathcal{G}(\cdot)$ to update the target dataset. We depict the high-confidence pseudo-labels provided by $\mathcal{G}(\cdot)$ as $\bar{y}_t$. The updated target dataset contains unlabeled data $\mathcal{T}_u = \{x_{tj}\}_{j=1}^{n_u}$ and pseudo-labeled data $\mathcal{T}_l = \{(x_{tj}, \bar{y}_{tj})\}_{j=1}^{n_l}$, which is described as:

$$\mathcal{T} = \mathcal{T}_u \cup \mathcal{T}_l, \quad (17)$$

where $n_u + n_l = n_t$, and we update $\mathcal{T}_u$ and $\mathcal{T}_l$ at the end of each episode. The pseudo-labeled target data will undergo self-supervised training in the subsequent episode, following the same training process as the source data with Eqs. (4) and (9). The last stage is the fine-tuning phase, where we train the labeled data ($\mathcal{S}$ and pseudo-labeled $\mathcal{T}_l$) using the following optimization formulation, which is also used in the pre-training phase:

$$\mathcal{L}_{labeled} = \mathcal{L}_{cls} + \mathcal{L}_{graph}. \quad (18)$$

Algorithm 1 in the Supp. Mat. summarizes the training process of CSCA.

Experiments

Datasets

We conduct experiments on four standard DA benchmarks: **Office-31** (Saenko et al. 2010), **Office-Home** (Venkateswara et al. 2017), **ImageCLEF-DA** (Caputo et al. 2014), and the very large scale **DomainNet** (Peng et al. 2019) (0.6 million images). More details about the datasets can be found in the Supp. Mat. For each dataset, we designate a subset (domain) of the dataset as the source domain and mix the remaining subsets to form a blended-target domain for constructing the transfer tasks, indicated as *source* → *rest*. For example, we take turns using each domain in the Office-31 dataset as the source domain to construct three transfer tasks: A→W/D, W→A/D, and D→A/W.

Methods	ImageCLEF-DA				
	B	C	I	P	Avg.
Source only	68.8	57.3	67.4	68.1	65.4
DAN	84.1	74.9	79.1	80.8	79.7
DANN	81.2	74.8	77.5	78.4	78.0
CDAN	86.2	74.7	79.2	82.4	80.6
SCDA	83.3	79.0	79.3	83.0	81.2
DCNT	75.7	76.6	77.3	81.6	77.8
AMDA	87.1	78.0	78.3	82.5	81.5
HTA	87.6	**81.3**	78.6	83.4	82.7
MCDA	86.6	77.9	78.6	83.5	81.6
CSCA(ours)	**87.9**	79.4	**80.2**	**83.8**	**82.8**

Table 2: Accuracy (%) of BTDA on the ImageCLEF-DA.

Implementation Details

To ensure a fair comparison with some relevant methods, we opt for the ResNet-50 (He et al. 2016) pre-trained on ImageNet (Russakovsky et al. 2015) as the backbone for all datasets. In line with (Yan et al. 2022), we adopt RandomFlip and RandomCrop as weak augmentation techniques, and RandAugment (Cubuk et al. 2020) as strong augmentation techniques. We compute the classification accuracies for all sub-target domains and use their average as the evaluation metric. More details about the network architecture and implementation details can be found in the Supp. Mat.

Methods	Ar	Cl	Pr	Rw	Avg.
CSCA (w/o $\mathcal{L}_{swd}$)	71.2	73.9	67.3	70.4	70.7
CSCA (w/o $\mathcal{L}_{graph}$)	66.2	68.7	64.6	65.9	66.4
CSCA (w/o $\mathcal{L}_{con}$)	71.7	72.1	66.8	68.9	69.9
CSCA	**73.5**	**75.0**	**69.7**	**73.6**	**73.0**

Table 3: Ablation studies of each component of CSCA.

Type	Amazon	Dslr	Webcam	Avg.
Only_P	93.6	86.6	88.5	89.6
Only_F	95.0	86.9	88.4	90.1
Double	**95.8**	**88.1**	**89.2**	**91.0**

Table 4: The effects of the prediction and feature aspects in the dual consistency regularization method. Each domain in the table represents the source domain, while the other sub-datasets constitute the blended-target domain.

Results

We assess our approach in comparison to various UDA methods, which include methods tailored for BTDA, such as AMEAN (Chen et al. 2019), CGCT (Roy et al. 2021), and MCDA (Xu, Wang, and Ling 2023). Due to the scarcity of existing BTDA methods, we also use MTDA methods for comparison: DCGCT (Roy et al. 2021), MTDA-PDT (Zhou et al. 2023), and MT-MTDA (Nguyen-Meidine et al. 2021). Additionally, we use other advanced UDA methods that can be extended and applied in the BTDA setting: AMDA (Wang et al. 2020), DAN (Long et al. 2015), CDAN (Long et al. 2018), DANN (Ganin and Lempitsky 2015), DCTN (Xu et al. 2018), HTA (Wu et al. 2023), JAN (Long et al. 2017), and SCDA (Li et al. 2021). The experimental **results on the DomainNet** dataset can be find in Supp. Mat.

Results on the Office-31. The left side of Tab. 1 displays the results on the Office-31 dataset. In the table, our method attains the highest average accuracy on the Office-31 dataset. Compared to the latest BTDA method, MCDA, our classification results in task A→W/D exceed those of MCDA by 3.4%. MCDA solely focuses on aligning category distributions, overlooking the semantic information embedded within the data. This shows the importance of simultaneously considering distribution alignment and semantic matching in enhancing the effectiveness of domain adaptation.

Results on the Office-Home. The right side of Tab. 1 shows the experimental results of our method on the Office-Home dataset. We can see that our method outperforms all compared methods on all the tasks of Office-Home, which shows the effectiveness and superiority of our method. Additionally, there was a notable 9% increase in average accuracy compared to the classical BTDA method, AMEAN. In contrast to the MTDA methods MTDA-PDT and MT-MTDA, our method remains superior because we propagate and learn semantic information across all sub-target domains rather than segregating each sub-target domain separately.

Results on the ImageCLEF-DA. Tab. 2 presents the classification results of the ImageCLEF-DA dataset. Our CSCA method exhibits exceptional performance and attains the highest average accuracy. It is worth highlighting that our method, trained solely in a single source domain, can achieve the desired performance across all sub-target domains. AMDA (Wang et al. 2020) is a multi-source multi-target domain adaptation method that uses multiple source domains to transfer knowledge for adaptation across multiple target domains. Although AMDA has explored complementary information between each source and target pair, it fails to harness the beneficial semantic correlation information among target domains, a factor crucial for enhancing model performance.

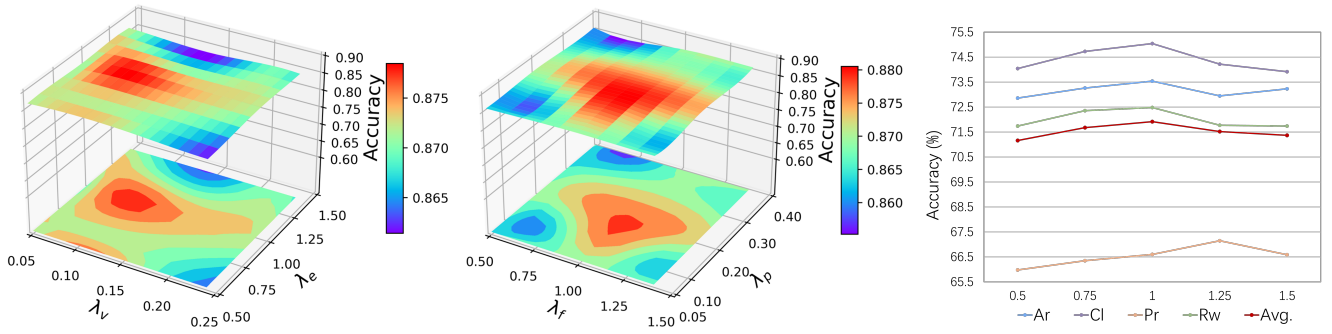
Analysis

Component-Wise Ablation. The ablation experiment results for each component of CSCA on the Office-Home dataset are presented in Tab. 3: 1) **CSCA (w/o $\mathcal{L}_{swd}$)** denotes the removal of loss $\mathcal{L}_{swd}$; 2) **CSCA (w/o $\mathcal{L}_{graph}$)** denotes the removal of loss $\mathcal{L}_{graph}$; and 3) **CSCA (w/o $\mathcal{L}_{con}$)** denotes the removal of loss $\mathcal{L}_{con}$. From the table, we can see that the semantic assignment method contributes the most to the performance of CSCA. However, without global distribution alignment, the model only achieved sub-optimal results, demonstrating the effectiveness of the collaboration between distribution alignment and semantic matching in achieving optimal adaptive performance. The results indicate that all proposed components contribute significantly to performance enhancement.

Effect of Double Consistency Regularization. To validate the effect of maintaining consistency in both prediction and feature aspects, we have presented results considering only the prediction aspect (referred to as "Only_P"), only the feature aspect (referred to as "Only_F"), and both aspects combined (referred to as "Double"). As shown in Tab. 4, it is clear that double consistency yields optimal results, which could facilitate the acquisition of domain-invariant and discriminative representations.

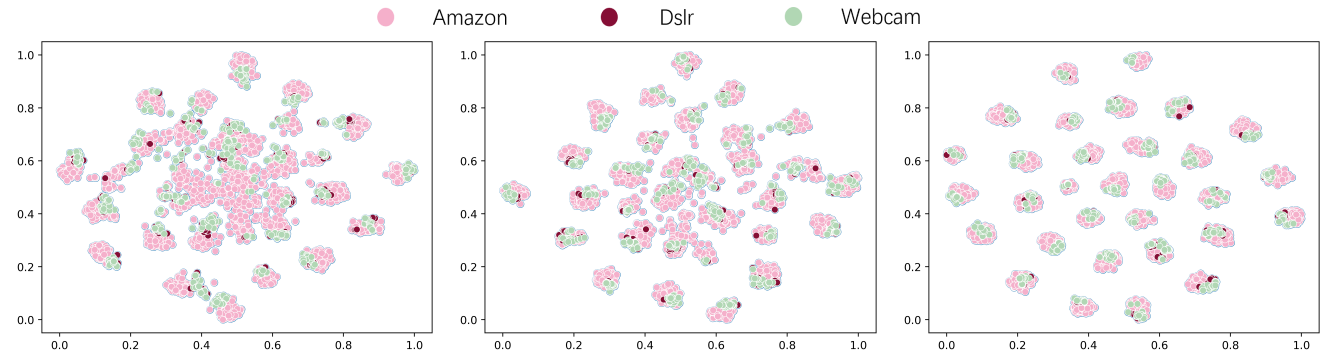
Hyper-Parameter Sensitivity Analysis. In this section, we perform hyper-parameter sensitivity analysis for our CSCA. Fig. 3 (a) shows that with balance parameters $\lambda_e = 1.0$ and $\lambda_v = 0.1$, our accuracy achieves its optimum. Fig. 3 (b) indicates that with $\lambda_p = 0.2$ and $\lambda_f = 1.0$, the model's performance reaches its peak. For the loss trade-off parameter λ_{swd}, we find that setting $\lambda_{swd} = 1.0$ yields the best results, as shown in Fig. 3 (c).

Visualization of Features. We have visualized the feature representations of several different methods by using the t-SNE technique (Selvaraju et al. 2017). Our experiments are performed on the Office-31 task A→D/W, as shown in Fig. 4. It is evident that the features in DANN and MCDA models lack well-constructed clusterings. In contrast, the features learned by our CSCA have formed more compact and discriminative clustering structures. This is because our learned features aggregate semantic information from similar samples across multiple domains, thus bringing similar features



(a) λ_v and λ_e (b) λ_f and λ_p (c) λ_{swd}

Figure 3: (a) is the hyper-parameter sensitivity analysis of λ_v and λ_e on task B→I/C/P (ImageCLEF-DA). (b) is the hyper-parameter sensitivity analysis of λ_f and λ_p on task B→I/C/P (ImageCLEF-DA). (c) is the sensitivity analysis of CSCA to parameters λ_{swd} on all Office-Home tasks.



(a) DANN (b) MCDA (c) CSCA

Figure 4: t-SNE feature visualization of DANN, MCDA, and CSCA (Ours) on the Office-31 with amazon as the source, i.e., A→D/W. Different colors represent different domains.

closer in the feature space.

Conclusion

In this paper, we propose a novel approach for a more challenging domain adaptation scenario, BTDA. Specifically, we minimize the distribution distance between the source and target to achieve inter-domain distribution alignment. Meanwhile, we design graph networks to conduct knowledge exchange and information sharing among all domains, which effectively alleviate the domain shifts in this complex hybrid data distribution. In our framework, domain alignment and semantic matching mutually complement each other to achieve better adaptation performance. Additionally, we propose a double consistency regularization method to reduce the interference of domain-specific information. Extensive experiment results on four benchmarks prove the efficacy of CSCA.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62176162) and the Guangdong Basic and Applied Basic Research Foundation (2023A1515012875, 2022A1515140099).

References

Bai, S.; Zhang, M.; Zhou, W.; Huang, S.; Luan, Z.; Wang, D.; and Chen, B. 2024. Prompt-based Distribution Alignment for Unsupervised Domain Adaptation. In *AAAI*, 729–737.

Caputo, B.; Müller, H.; Martinez-Gomez, J.; Villegas, M.; Acar, B.; Patricia, N.; Marvasti, N.; Üsküdarl, S.; Paredes, R.; Cazorla, M.; Garcia-Varea, I.; and Morell, V. 2014. ImageCLEF 2014: Overview and Analysis of the Results. *Lect. Notes Comput. Sci.*, 192–211.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 1597–1607.

Chen, Z.; Zhuang, J.; Liang, X.; and Lin, L. 2019. Blending-Target Domain Adaptation by Adversarial Meta-Adaptation Networks. In *CVPR*, 2248–2257.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical Automated Data Augmentation With a Reduced Search Space. In *CVPR*, 702–703.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised Domain Adaptation by Back-Propagation. In *ICML*, 1180–1189.

Han, Z.; Sun, H.; and Yin, Y. 2022. Learning Transferable Parameters for Unsupervised Domain Adaptation. *IEEE TIP*, 31(6): 6424–6439.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Isobe, T.; Jia, X.; Chen, S.; He, J.; Shi, Y.; Liu, J.; Lu, H.; and Wang, S. 2021. Multi-Target Domain Adaptation with Collaborative Consistency Learning. In *CVPR*, 8187–8196.

Jing, M.; Zhen, X.; Li, J.; and Snoek, C. G. M. 2023. Order-preserving Consistency Regularization for Domain Adaptation and Generalization. In *ICCV*, 18916–18927.

Kiran, M.; Nguyen-Meidine, L. T.; Pedersoli, M.; Dolz, J.; Blais-Morin, L.-A.; and Granger, E. 2022. Incremental Multi-Target Domain Adaptation for Object Detection with Efficient Domain Transfer. *Pattern Recognit.*, 634: 140–156.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial Machine Learning at Scale. In *ICLR*.

Lee, C.-Y.; Batra, T.; Baig, M. H.; and Ulbricht, D. 2019. Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation. In *CVPR*, 10285–10295.

Li, S.; Xie, M.; Lv, F.; Liu, C. H.; Liang, J.; Qin, C.; and Li, W. 2021. Semantic Concentration for Domain Adaptation. In *CVPR*, 9102–9111.

Liang, J.; Hu, D.; and Feng, J. 2020. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. In *ICML*, 6028–6039.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Analysis of Representations for Domain Adaptation. In *ICML*, 97–105.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional Adversarial Domain Adaptation. In *NeurIPS*, 1640–1650.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*, 2208–2217.

Ma, X.; Zhang, T.; and Xu, C. 2019. GCAN: Graph Convolutional Adversarial Network for Unsupervised Domain Adaptation. In *CVPR*, 8266–8276.

Ngo, B. H.; Chae, Y. J.; Park, J. H.; Kim, J. H.; and Cho, S. I. 2023. Easy-to-Hard Structure for Remote Sensing Scene Classification in Multitarget Domain Adaptation. *IEEE TGRS*, 61: 1–15.

Nguyen-Meidine, L. T.; Belal, A.; Kiran, M.; Dolz, J.; Blais-Morin, L.-A.; and Granger, E. 2021. Unsupervised Multi-Target Domain Adaptation Through Knowledge Distillation. In *WACV*, 1339–1347.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *ICCV*, 1406–1415.

Roy, S.; Krivosheev, E.; Zhong, Z.; Sebe, N.; and Ricci, E. 2021. Curriculum Graph Co-Teaching for Multi-Target Domain Adaptation. In *CVPR*, 5351–5360.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; and Karpathy, A. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115: 211–252.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting Visual Category Models to New Domains. In *ECCV*, 213–226.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *ICCV*, 618–626.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *NeurIPS*, 596–608.

Tian, Q.; Ma, C.; Cao, M.; Wan, J.; Lei, Z.; and Chen, S. 2022. Unsupervised Multitarget Domain Adaptation With Dictionary-Bridged Knowledge Exploitation. *IEEE TNNLS*, 35: 3464–3477.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *CVPR*, 5018–5027.

Wang, Y.; Zhang, Z.; Hao, W.; and Song, C. 2020. Attention Guided Multiple Source and Target Domain Adaptation. *IEEE TIP*, 30: 892–906.

Wiles, O.; Gowal, S.; Stimberg, F.; Alvise-Rebuffi, S.; Ira Ktena, K. D.; and Cemgil, T. 2021. A Fine-Grained Analysis on Distribution Shift. In *ICLR*.

Wu, Z.; Meng, M.; Liang, T.; and Wu, J. 2023. Hierarchical Triple-Level Alignment for Multiple Source and Target Domain Adaptation. *Appl. Intell.*, 54(4): 3766–3782.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE TNNLS*, 32(1): 4–24.

Xie, B.; Li, S.; Lv, F.; Liu, C. H.; Wang, G.; and Wu, D. 2022. A Collaborative Alignment Framework of Transferable Knowledge Extraction for Unsupervised Domain Adaptation. *IEEE TKDE*, 35: 6518–6533.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. In *NeurIPS*, 6256–6268.

Xu, P.; Wang, B.; and Ling, C. 2023. Class Overwhelms: Mutual Conditional Blended-Target Domain Adaptation. In *AAAI*, 3, 3036–3044.

Xu, R.; Chen, Z.; Zuo, W.; Yan, J.; and Lin, L. 2018. Deep Cocktail Network: Multi-Source Unsupervised Domain Adaptation with Category Shift. In *CVPR*, 3964–3973.

Yan, Z.; Wu, Y.; Li, G.; Qin, Y.; Han, X.; and Cui, S. 2022. Multi-level Consistency Learning for Semi-supervised Domain Adaptation. In *IJCAI*.

Yang, X.; Deng, C.; Liu, T.; and Tao, D. 2020. Heterogeneous Graph Attention Network for Unsupervised Multiple-Target Domain Adaptation. *IEEE TPAMI*, 44(4): 1992–2003.

Yuan, J.; Hou, F.; Du, Y.; Shi, Z.; Geng, X.; Fan, J.; and Rui, Y. 2022. Self-Supervised Graph Neural Network for Multi-Source Domain Adaptation. In *ACMMM*, 3907–3916.

Zhang, Y.; Li, W.; Zhang, M.; Wang, S.; Tao, R.; and Du, Q. 2024. Graph Information Aggregation Cross-Domain Few-Shot Learning for Hyperspectral Image Classification. *IEEE TNNLS*, 35(2): 1912–1925.

Zhou; Jiazhong; Tian, Q.; and Lu, Z. 2023. Progressive Decoupled Target-into-Source Multi-Target Domain Adaptation. *Inf. Sci.*, 634.