Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2963195, 12 pages
https://doi.org/10.1155/2022/2963195

Research Article

Classifier Adaptation Based on Modified Label Propagation for Unsupervised Domain Adaptation

Yongjie Du,[1] **Deyun Zhou**,[1] **Yu Xie**,[2] **Weiren Kong**,[1] **Xiaoyang Li**,[1] **Jiao Shi**,[1] **and Yu Lei**[1]

[1]*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi Province 710071, China*
[2]*Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China*

Correspondence should be addressed to Deyun Zhou; dyzhounpu@nwpu.edu.cn

Received 9 June 2022; Revised 4 July 2022; Accepted 7 July 2022; Published 9 August 2022

Academic Editor: Wentao Li

Unsupervised domain adaptation endeavors to learn a desirable classifier for a target domain by transferring knowledge learned from a related (source) domain. Existing approaches focus on deriving domain-invariant feature representations by aligning the domain distributions. However, those works often require an extra classifier. In contrast, this paper proposes a classifier adaptation method based on modified label propagation (CAMLP) for unsupervised domain adaptation. Inspired by pseudolabeling, CAMLP proposes a simple but effective measurement for relationships among cross-domain samples. Thus, samples from distinct domains are constructed in a same graph. The true labels can then propagate from the source domain to the target one along the graph. We also propose a consistency-aware pseudolabel annotation to alleviate the problem of negative transfer caused by unreliable pseudo labels. Extensive experiments on several benchmark datasets confirm that the proposed method performs favorably against the state-of-the-art approaches.

1. Introduction

Domain adaptation [1–3] transfers knowledge from a label-well source domain to the label-scarce target domain, and there is usually a certain degree of data distribution discrepancy between the source and target domains. Domain adaptation has a wide range of applications in medical diagnosis [4, 5], intelligent traffic [6–8], and other IOT (Internet of Things) fields [9, 10]. For example, domain adaptation is widely used in image semantic segmentation tasks to alleviate the huge cost of dense annotation [11–14]. And it is also applied to strengthen the generalization of person reidentification models [6, 15–17]. In fact, machine learning has made a amazing progress in these fields recently, but this relies on vast amounts of label-accurate training data and requires same distribution of training data (source domain) and test data (target domain). In real world scenarios, however, annotating samples for every scenario is uneconomical or even impossible. Domain adaptation tackles this problem by applying the source-domain-learned ability to the target domain. Experience on riding a bike, for example, is helpful for riding a motorcycle. Due to the distribution shift (discrepancy), the model performance will deteriorate when applied directly from one domain (source domain) to another (target domain). Domain adaptation mitigates this effect through bridging the gaps between domains. In general, domain adaptation is mainly classified into semi-supervised domain adaptation (SSDA) [18–20] with a few labels of target samples being available, unsupervised domain adaptation (UDA) [21] without any target label. The SSDA uses the limited labeled target samples to facilitate the model training with the label-rich source data. Therefore, the SSDA-learned classifier usually has a more easily distinguishable classification boundary in the target domain than the UDA-learned one [22].

In this study, we focus mainly on the UDA, which is more challenging and practical. The key of UDA is how to exploit common knowledge across domains. Existing

approaches generally aligned distinct domains by projection or transformation, such as searching common latent feature representations in Hilbert space [23–25]. Maximum mean discrepancy (MMD) is a classic and efficient one of these strategies. The structural (geometric) information of samples has recently attracted the interest of researchers, which can efficiently characterize the relationship between samples, according to certain relevant research [26–28]. By investigating the underlying structure across all (source and target) samples, the graph-based technique offers a high-efficiency way to develop a classifier [26]. All approaches above align the domain distribution but require an additional classifier. In contrast, this paper learns a domain-invariant classifier directly.

To this end, a novel UDA method, classifier adaptation based on modified label propagation (CAMLP), is proposed by exploiting cross-domain and within-domain structural information. Samples from different domains are constructed into a k-nearest neighbor graph. The nodes of the graph represent the observations in the source or target domain. The edges of the graph indicate the relationships between these observations, which are used to be measured in traditional ways, such as cosine similarity. However, the traditional similarity metrics may fail to accurately reflect the affinity relationships between cross-domain samples due to domain discrepancy. Our modified label propagation (MLP) proposes to measure these relationship by pseudo labeling, simply yet effectively. Then, the true labels transfer from the source domain to the target one along the graph. In addition, we propose a consistency-aware pseudolabel annotation (CPLA) to alleviate the negative transfers caused by incorrect labels. In conclusion, the contributions of this study are as follows:

(i) A unsupervised domain adaptation method, classifier adaptation based on modified label propagation (CAMLP), is proposed through exploiting structural information and pseudo labeling

(ii) A simple yet effective approach is proposed to measure the relationship between cross-domain samples. Thus, samples of different domains can be constructed into the same graph

(iii) A consistency-aware pseudolabel annotation method is presented, which significantly alleviates the impact of incorrect labels

(iv) Extensive experiments on four benchmark datasets demonstrate that our method outperforms the mainstream domain adaptation methods

The rest of this paper is organized as follows. Section 2 introduces relevant topics of unsupervised domain adaptation and some basics of this work. Section 3 delves into the specifics of our proposed method. Sections 4 and 5 provide extensive experiments and discussion. Finally, our work is concluded in Section 6.

2. Related Work

2.1. Unsupervised Domain Adaptation. Mainstream unsupervised domain adaptation methods learn a domain-invariant feature representation by narrowing the domain discrepancy. These methods can align the domain distribution well but may damage the category discrimination within the target domain. Accordingly, there are two important challenges in UDA: aligning the domain distribution at the class level and maintaining the category differentiability within the domain. To this end, two important tools are proposed: the pseudolabeling and structural exploitation for UDA.

The pseudolabeling was commonly utilized to reduce domain discrepancy at the category level, which helped to efficiently apply classifiers learned from the source domain to the target one [29]. Pseudolabels were often generated by a source-domain-trained classifier [25, 30]. Due to domain shift, however, these pseudolabels may contain incorrect ones, which would cause the negative transfer problem. Pei et al. [31] alleviated the problem by the soft labeling method. Refs. [28, 32, 33] proposed the confidence-aware label filtering method, which only allowed the high-confidence pseudolabels to narrow domain discrepancy. PACET [34] effectively reduced the uncertainty in pseudo labels by filtering the target samples progressively and describing relations among samples with class confidence scores.

UDA differs from unsupervised learning mainly in the availability of unlabeled test (target) data. The data can be regarded as coming from a distribution supported by a low-dimensional manifold embedded in a high-dimensional space in terms of geometric structure [35]. In a sense, closed samples intend to belong to the same class. The underlying structure within the target domain helps distinguish the different categories, so that it is often used as an auxiliary knowledge to enhance domain adaptive methods. For example, GTL [27] managed UDA by optimizing the empirical likelihood while maintaining the geometric structure at the same time. JDA [23], LSC [36] and GAKT [26] exploited the structure in the target domain to increase the confidence of pseudolabels. For better aligning the conditional distribution between two domains, CCSL [29] used both the label and structural information within and across domains.

In contrast, this paper proposes a classifier adaptive method-based modified label propagation (CAMLP) for unsupervised domain adaptation. GTL [27] and SPL [28] are the most similar to our method. In several ways, CAMLP differs significantly from them. To begin with, GTL optimizes both empirical likelihood and geometric structure concurrently to tackle domain adaptation tasks, but CAMLP keeps the geometric structure while aligning the conditional distribution. Both CAMLP and SPL employ pseudolabeling to reduce the conditional distribution discrepancy. CAMLP immediately learns a domain-invariant classifier, whereas SPL tries to learn a domain-invariant feature representation.

2.2. Label Propagation. Graphs are widely used to model structured and relational data. The label propagation is a

Input: $\mathbf{X}$, cluster number C.
Output: $\widehat{\mathbf{Y}}^{clu}$.
1 Construct the affinity matrix $\mathbf{W}$ from $\mathbf{X}$;
2 Calculate the degree matrix: $\mathbf{D} = \text{diag}\,(d_1, d_2, \cdots, d_n)$, $d_i = \sum_{j=1}^{n}\{w\}_{ij}$;
3 Calculate Laplacian matrix: $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
4 Normalize Laplacian matrix: $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1}\mathbf{L}$;
5 $\mathbf{V}$ = eigenvector$(\mathbf{L}_{\text{norm}})$, $\mathbf{V} \in \mathbb{R}^{n\times C}$;
6 Cluster $\mathbf{V}$ into C categories by k-means clustering;
7 Obtain clustering labels $\widehat{\mathbf{Y}}^{clu} \in \mathbb{R}^{n\times 1}$;

Algorithm 1: spectralCluster.

classifying function that is sufficiently smooth to inherent the structure disclosed by available labeled and unlabeled graph nodes. Given a graph $G = (\mathbf{X}, \mathbf{W})$, where $\mathbf{X}$ represents the graph nodes and $\mathbf{W}$ indicates graph edges. $\mathbf{X} = [\mathbf{X}^L; \mathbf{X}^U]$ and the labels of $\mathbf{X}^L$ are available ($\mathbf{Y}^L$) whereas the labels of $\mathbf{X}^U$ are not. $\mathbf{W}$ represents the affinity matrix of $\mathbf{X}$, whose element w_{ij} represents the relationship between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. Label propagation is able to propagate the labels from $\mathbf{X}^L$ to $\mathbf{X}^U$ by learning the indicated matrix $\mathbf{F}$. Specifically, $\mathbf{F}$ is computed by minimizing the following cost function:

$$Q(\mathbf{F}) = 12\sum_{i,j=1}^{l+u} w_{ij}\left\|1\sqrt{d_i}\mathbf{F}_i - 1\sqrt{d_j}\mathbf{F}_j\right\|^2 + 12\mu\sum_{i=1}^{l}\|\mathbf{F}_i - \mathbf{Y}_i\|^2, \tag{1}$$

where $l = |\mathbf{X}^L|$ and $u = |\mathbf{X}^U|$ are the corresponding node numbers. The first element on the right causes neighboring samples to have similar indicated vectors $\mathbf{F}_*$, whereas the second guarantees the prediction close to their ground truth labels. The trade-off between these two constraints is captured by the hyper-parameter μ. $\mathbf{Y}_*$ indicates the one-hot vector of the corresponding ground truth label. Proved by [37], the optimal $\mathbf{F}$ is computed by the following equations:

$$\begin{aligned}\mathbf{F} &= (1-\alpha)(\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{Y},\\ \mathbf{S} &= \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2},\\ \mathbf{D} &= \text{diag}\,(d_1, d_2, \cdots, d_n), d_i = \sum_{j=1}^{n} w_{ij},\end{aligned} \tag{2}$$

where $\alpha = 11 + \mu$, $\mathbf{S}$ is the propagating matrix, and $\mathbf{D}$ is the degree matrix of $\mathbf{W}$. $\mathbf{Y}$ is the one-hot matrix with $Y_{ij} = 1$ if $\mathbf{x}_i$ is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise. Then, $\mathbf{x}_i$ can be predicted by the following:

$$\widehat{y}(\mathbf{x}_i) = \underset{1\le j\le c}{\text{argmax}} F'_{ij}. \tag{3}$$

In summary, the affinity matrix $\mathbf{W}$ and matrix $\mathbf{Y}$ may identify the labels of nodes uniquely.

2.3. Spectral Clustering. Spectral clustering (SC) [38] is a classical unsupervised learning method that can split unlabeled data into several groups. SC firstly projects the training data into a low-dimensional space, then groups them by K-means or K-medoids. The dimension of the low-dimensional space is based on the eigenvectors of the Laplacian matrix about $\mathbf{W}$. And $\mathbf{W}$ is the affinity matrix describing the relation between samples. The specific implementation process of spectral clustering is shown in Algorithm 1. $\mathbf{W}$ is constructed by calculating the similarity between samples. Note that Laplacian matrix $\mathbf{L}$ is normalized by the Shi-Malik method [39] in Line 4.

3. Methodology

In this section, we offer the problem formulation first and then introduce the suggested method: classifier adaptation based on modified label propagation (CAMLP) for unsupervised domain adaptation in detail.

3.1. Problem Formulation. This work focuses on UDA with label-well source domain ($\mathscr{D}_s$) and label-scarce target domain ($\mathscr{D}_t$). Specifically, the labeled data $\{\mathbf{X}^s, \mathbf{Y}^s\} \in \mathscr{D}_s$, unlabeled data $\{\mathbf{X}^t\} \in \mathscr{D}_t$. $\mathscr{X}_s$ and $\mathscr{Y}_s$ ($\mathscr{X}_t$ and $\mathscr{Y}_t$) represent the feature space and label space of the source (target) domain, respectively. Our work focuses the scenario: label space $\mathrm{Y}_s = \mathrm{Y}_t$, whereas the marginal probability $P_s(\mathscr{X}_s) \neq P_t(\mathscr{X}_t)$ and conditional probability $Q_s(\mathscr{X}_s|\mathscr{Y}_s) \neq Q_t(\mathscr{X}_t|\mathscr{Y}_t)$. The purpose of UDA is to learn a classifier f to accurately predict the target samples, i.e., $f(\mathbf{X}^s, \mathbf{Y}^s, \mathbf{X}^t) \longrightarrow \widehat{\mathbf{Y}}^t$. Table 1 shows the primary notations and descriptions used in this study.

3.2. The Overall Framework. In this paper, we proposed a classifier adaptation-based modified label propagation (CAMLP) for unsupervised domain adaptation, as illustrated in Figure 1. To begin with, a k-nearest neighbor (knn) graph is constructed for two distinct domains. The labels are then transferred from the source domain to the target one via the modified label propagation. When constructing graphs, it is difficult to accurately measure the affinity relationship between cross-domain data only by conventional methods, due to domain discrepancy. Inspired by the pseudolabeling technique, our modified label propagation presents a novel way for solving this challenge. In this

Table 1: Notations and descriptions.

Notation	Description
	Source/target domain
$\mathbf{X}^s/\mathbf{X}^t$	Source/target sample matrix
$\mathbf{Y}^s/\mathbf{Y}^t$	Source/target label matrix
n_s/n_t	Number of source/target samples
p_1/p_2	Nearest neighbor number in LP/SC
d/d_2	Feature dimension before/after PCA
$\mathbf{W}$	Affinity matrix

method, the accuracy of pseudolabels is crucial, because misclassified samples frequently degrade performance. To this end, this work presents a consistency-aware pseudolabel annotation method. The idea is that samples in the same cluster are more likely to belong to the same class, according to the clustering assumption. The structure within the target domain helps to obtain more accurate predicted labels. Our approach is outlined in depth in the following subsections.

3.3. Graph Construction. Let $\mathbf{X} = [\mathbf{X}^s; \mathbf{X}^t]^T \in \mathbb{R}^{n\times d}$, $\mathbf{Y} = [\mathbf{Y}^s; \mathbf{0}]^T \in \mathbb{R}^{n\times c}$ where $n = n_s + n_t$, $n_s = |\mathbf{X}^s|$, $n_t = |\mathbf{X}^t|$. In high-dimensional data, there is information redundancy increasing the computational cost. We use principal component analysis (PCA) to reduce the dimension of data, similar to many other UDA approaches [23, 28], i.e., $\mathrm{PCA}(\mathbf{X}) \longrightarrow \tilde{\mathbf{X}}$. The feature vector in $\tilde{\mathbf{X}}$ is then subjected to L2 normalization, which compels samples from the two domains to be dispersed on the surface of same hypersphere. During the domain adaptation technique, we retain the inherent data distributions of the two domains by preserving the geometric structure. To properly depict the structural relationship in feature space, we construct a knn graph $\mathbf{G} = (\tilde{\mathbf{X}}, \mathbf{W})$ with n vertexes, where $\mathbf{W}$ is the affinity matrix of $\tilde{\mathbf{X}}$ and its entries on the diagonal are 0.

3.4. Transfer Labels via Modified Label Propagation. Calculating the affinity matrix W is critical for labeling propagation, according to Section 2.2. W is represented as a partitioned matrix in the following way for convenience:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{ss} & \mathbf{W}_{st} \\ \mathbf{W}_{st}^T & \mathbf{W}_{tt} \end{bmatrix}, \tag{4}$$

where $\mathbf{W}_{ss}$ and $\mathbf{W}_{tt}$ denote the relationships within the source and target domains, respectively; $\mathbf{W}_{st}$ denotes the relationships between cross-domain samples. Surely, $\mathbf{W}_{st}$ is difficult to measure precisely by the traditional way because of domain discrepancy. To tackle this issue, we propose a label-aware representation of relationship between cross-domain samples. Specifically, we firstly annotate target samples. Then, the relationship between two samples is 1 if their labels are the same, and 0 otherwise. In a nutshell, $\mathbf{W}_{st}$ and $\mathbf{W}_{ss}$ are calculated by the following equation:

$$w_{ij} = \begin{cases} 1, & \left(y(x_i) = \widehat{y}\left(x_j\right)\right) \cap (i \neq j) \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$
$$s.t. \left(1 \le i \le n_s, 1 + n_s \le j \le n\right) \cup \left(1 \le i, j \le n_s\right)$$

where $\widehat{y}(*)$ and $y(*)$ denote the pseudolabel and the ground truth label, respectively. $\mathbf{W}_{tt}$ is calculated through the Euclidean distance, as demonstrated in the following:

$$\mathrm{Dist}_{ij} = \begin{cases} \mathrm{dist} < x_i, x_j >, & x_i \in \mathcal{N}_p\left(x_j\right), x_j \in \mathcal{N}_p(x_i) \\ 0, & \text{otherwise} \end{cases}$$
$$w_{ij} = e^{-\left(\mathrm{Dist}_{ij}\right)^2}$$
$$s.t. n_s + 1 \le i, j \le n \tag{6}$$

where $\mathcal{N}_p(x_j)$, $\mathcal{N}_p(x_i)$ are the p nearest neighbor set of x_j and x_i, respectively. According to Section 2.2, the target labels $\widehat{\mathbf{Y}}^t$ are obtained via the original label propagation, as shown as Algorithm 2. It is worth noting that the incorrect ones of pseudo labels may damage the model (result in negative transfer problem). To solve this problem, we propose a consistency-aware strategy to correct these wrong labels.

3.5. Consistency-Aware Pseudolabel Annotating. The existing pseudolabel initialization schemes are entirely dependent on the source domain. In contrast, our method exploits the knowledge of the source domain and the target-domain structural information together to predict target samples. We construct a new knn graph for the target sample and then split the graph into C subgraphs through spectral clustering. In other words, the target samples are grouped into C clusters. A ballot-aware alignment method is proposed to match these cluster labels with the known categories. Specifically, a source-domain-learned classifier predicts the initial target-domain pseudolabels $\widehat{\mathbf{Y}}^{t''}$. The drop points of $\widehat{\mathbf{Y}}^{t''}$ on each target cluster are calculated. We define that i is the category label for a certain cluster, if there are the most points belonging to i in the cluster according to the calculated drop-points. Therefore, the new pseudolabels $\widehat{\mathbf{Y}}^{sc}$ are obtained. At this point, $\widehat{\mathbf{Y}}^{t''}$ contains the source-domain knowledge, and $\widehat{\mathbf{Y}}^{sc}$ contains target-domain underlying structure. Based on this, we propose a consistency-aware strategy: only reserving the same elements in $\widehat{\mathbf{Y}}^{t''}$ and $\widehat{\mathbf{Y}}^{sc}$ as pseudo labels.

The specific implementation process of the method is shown in Algorithm 3. In Lines 1-2, a source-domain-learned classifier predicts the target samples. In Line 3, the spectral clustering groups target samples into C clusters. In Lines 4-9, the target clusters are matched with the known categories, where $\mathfrak{n}_{ij}$ interprets the number of pseudolabels

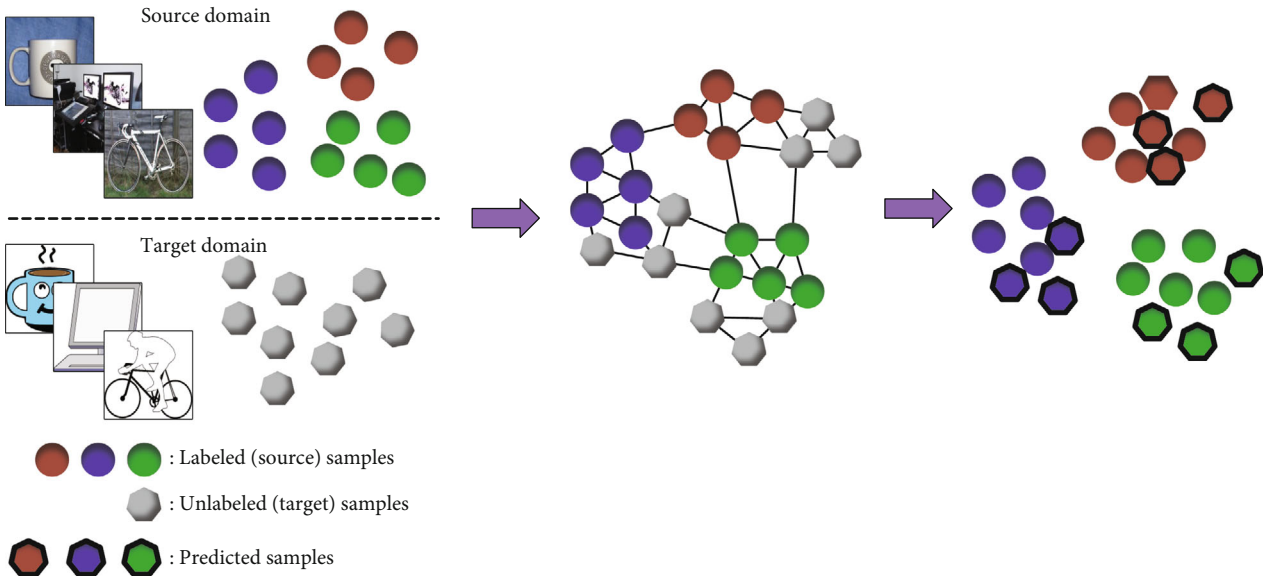

Figure 1: Illustration of the proposed approach (best viewed in color). The distinct domain samples are constructed into the same graph. The ground truth labels then transfer from source domain to target one along the graph.

Input: $\tilde{\mathbf{X}}^s$, $\mathbf{Y}^s$, $\tilde{\mathbf{X}}^t$, $\hat{\mathbf{Y}}^{t'}$, class num C, hyperparameter α
Output: $\hat{\mathbf{Y}}^t$.
1 Construct the affinity matrix: $\mathbf{W}$ via equations (5) and (6);
2 Construct one-hot-label matrix: $\mathbf{Y} = [\mathbf{Y}^s; 0]^{n\times C}$;
3 Calculate $\mathbf{D} = \text{diag}(d_1, d_2, \cdots, d_n)$, $d_i = \sum_{j=1}^{n} w_{ij}$;
4 $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$;
5 $\mathbf{F} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{Y}$;
6 $\hat{y}_i = \underset{1\le j\le C}{\text{argmax}}(\mathbf{F}_{ij})$;
7 Return the predicted labels: $\hat{\mathbf{Y}}^t = \{\hat{y}_i\}_{1\le i\le n}$.

Algorithm 2: Prediction via MLP.

Input: $\tilde{\mathbf{X}}^s$, $\mathbf{Y}^s$, $\tilde{\mathbf{X}}^t$.
Output: pseudolabels $\hat{\mathbf{Y}}^{t'}$.
1 Learn a classifier $f : f(\tilde{\mathbf{X}}^s, \mathbf{Y}^s, \tilde{\mathbf{X}}^t) \longrightarrow \hat{\mathbf{Y}}^t$;
2 $\hat{\mathbf{Y}}^{t''} \longleftarrow f(\tilde{\mathbf{X}}^t)$;
3 $\hat{\mathbf{Y}}^{\text{clu}} \longleftarrow \text{spectralCluster}(\tilde{\mathbf{X}}^t)$;
4 $\hat{\mathbf{Y}}^{sc} = [0]^{n_t\times 1}$;
5 **for** i =1: C **do**
6 Calculate $\{\mathbf{n}_{ij}\}_{1\le j\le c}$;
7 $\hat{y} = \underset{1\le j\le c}{\text{argmax}}(\mathbf{n}_{ij})$;
8 $\hat{\mathbf{Y}}^{sc}(\hat{\mathbf{Y}}^{\text{clu}} == i) = \hat{y}$;
9 end.
10 $\hat{\mathbf{Y}}^{t'} = \{\hat{y}^{t'}{}_i\}_{1\le i\le n_t}$, $\hat{y}_i^{t'} = \begin{cases} \hat{y}_i^{t''}, & \hat{y}_i^{t''} = \hat{y}_i^{sc} \\ 0, & \text{otherwise} \end{cases}$

Algorithm 3: Consistency-aware pseudolabel annotation.

equaling i in the jth cluster. Line 10 shows the consistency-aware label filtering strategy. It is worth noting that an iterative framework is utilized. Specifically, during each training session, the prediction of target samples are used as the pseudo labels for the next iteration. An incremental training strategy is used where the pseudo labels with high confidence are fixed in each iteration, and only the remaining pseudo labels are updated in subsequent iterations.

4. Experiments

Firstly, we elaborate the benchmark datasets and the experimental setup in this part. The results of comparisons with baseline domain adaptation techniques are then provided and analyzed. Finally, we test the sensitivity of various parameters and give the visualization of the confusion matrix.

4.1. Datasets and Evaluation Criteria. **Office-Caltech10** [40] is a popular dataset in the object recognition area, which

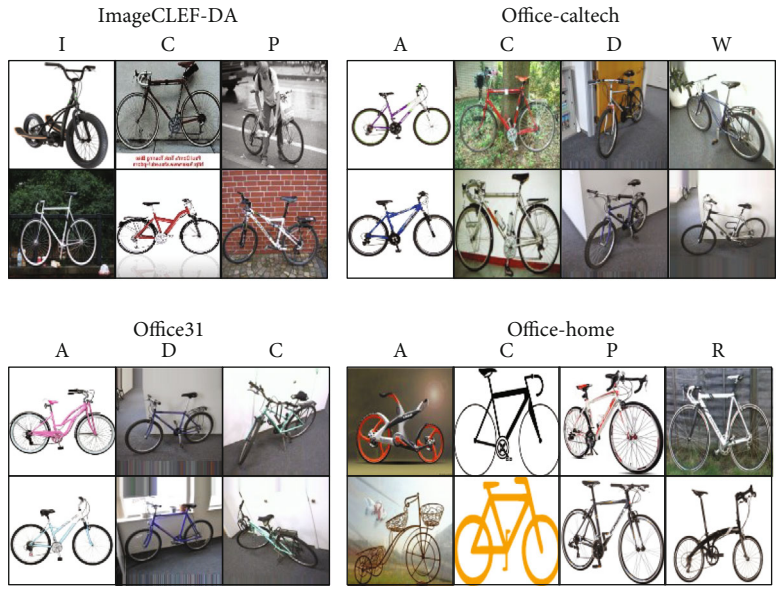

Figure 2: Objects in different domain for four datasets.

contains 2533 samples across four domains: **Amazon** (**A**), **Caltech** (**C**), **Webcam** (**W**), and **DSLR** (**D**). Each domain contains 10 object classes, such as backpacks, bicycles, and calculators.

ImageCLEF-DA is a benchmark dataset for ImageCLEF 2014 Domain Adaptation Challenge https://www.imageclef.org/2014/adaptation, which are composed of 1800 samples across three domains: **Caltech** (**C**), **Imagenet** (**I**), and **PASCAL** (**P**). Each domain provides 12 classes of objects, including airplane, bike, and people.

Office31 [41] is a medium-size dataset for the object recognition, which has 4110 images in 31 categories. These images are divided into three domains: **Amazon** (**A**), **DSLR** (**B**), and **Webcam** (**W**) according to their source. Specifically, **A** contains 2817 images with clean background and unified scale; **B** has 498 low-noise high-resolution pictures (4288×2848); **W** are composed of 795 low-resolution images with flaws, such as significant noise, color, and white balance artifacts.

Office-Home [42] is a larger dataset than **Office31** in object recognition area, which includes 15588 images in 65 categories. There are four domains: **Art** (**A**), **Clipart** (**C**), **Product** (**P**), and **Real-word** (**R**). Specifically, **A** has 2427 images in the form of sketches, paintings, etc.; **C** contains 4365 clipart-images; **P** consists of the set with 4439 clean-background pictures of products; **R** includes 15500 images in real world.

These four benchmark datasets are widely utilized in UDA and some samples of them are shown in Figure 2. Decaf6 features [43] of **Office-Caltech10** and Resnet50 features [44] of others datasets are utilized in our experiments. The information of four datasets is shown in Table 2. In terms of every dataset, subtasks are performed by training in one domain and testing in another. For example, **C** $\longrightarrow$ **I** indicates training on the domain **Caltech** while testing on the domain **Imagenet**. Similar to works [27, 28, 45], the predicted accuracy of target samples is utilized as the assessment criteria. The accuracy is computed by the following equation:

Table 2: Statistics of datasets.

Dataset	#Feature	#Domain	#Object	#Sample
Office-caltech10	4096	4	10	2533
ImageCLEF-DA	2048	3	12	2391
Office31	2048	3	31	4110
Office-Home	2048	4	65	15500

$$\text{Accurary} = |\{x : x \in D_t \wedge (f(x) = y(x))\}| / |\{x : x \in D_t\}|, \quad (7)$$

where $f(*)$ and $y(*)$ represent predicted label and the ground truth label, respectively.

4.2. Environment. All our experiments ran on a same desktop-computer with a single CPU (Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz), 16.0 GB RAM, and 256 G SSD. The programming language is MATLAB.

4.3. Experimental Setting. The label propagation and spectral clustering both employ the k-nearest neighbor graph, however, the neighbor-node number differs between them, $p_1 = 15$ for the former, $p_2 = 20$ for the latter. The dimension of PCA space is $d_1 = 256$. And the label propagation parameter α is empirically set as 0.9 [37]. In the succeeding subsections, the impact of these hyperparameters on model will be examined in detail.

4.4. Baseline Methods. Our strategy is compared to a nonadaptation approach as well as several standard methodologies to assess performance. The follows are a brief introduction of these comparative approaches: the **1NN** [46] is a baseline method without adaptation. Specifically, a

Table 3: Accuracy (%) on ImageCLEF-DA.

Method	1NN	TCA	JDA	GFK	MAFS	JPDA	CRCCB	CAMLP
C⟶I	85.16	86.67	90.83	84.67	89.83	84.00	89.83	**93.70 ± 0.07**
C⟶P	69.16	72.08	73.43	70.39	72.83	73.60	71.91	**76.35 ± 1.06**
I⟶C	91.16	93.33	93.67	93.00	93.17	86.83	94.16	**94.67 ± 0.00**
I⟶P	73.16	75.63	76.48	76.14	76.83	76.31	**78.84**	77.70 ± 0.08
P⟶C	81.33	85.33	83.83	86.00	85.33	80.67	**94.50**	94.33 ± 0.00
P⟶I	74.50	79.17	78.50	80.00	80.83	79.17	90.83	**92.30 ± 0.07**
Aver	79.08	82.04	82.79	81.70	83.14	80.10	86.68	**88.17 ± 0.20**

Table 4: Accuracy (%) on Office-Caltech10.

Method	1NN	TCA	JDA	GFK	MAFS	JPDA	**CAMLP**
A⟶C	70.34	84.33	84.68	79.52	87.36	73.82	**89.49 ± 0.00**
A⟶D	64.96	83.44	81.53	80.89	**86.62**	84.71	84.2 ± 1.59
A⟶W	57.28	74.92	76.27	68.47	81.02	72.88	**85.08 ± 1.44**
C⟶A	85.69	89.35	90.40	87.79	90.81	84.76	**92.17 ± 0.00**
C⟶D	74.52	84.71	87.26	87.26	89.81	86.62	**92.99 ± 0.78**
C⟶W	66.10	82.03	86.44	80.34	87.46	81.02	**88.34 ± 1.11**
D⟶A	62.73	88.62	91.02	86.43	90.40	86.53	**91.96 ± 0.00**
D⟶C	52.09	80.50	84.59	77.56	85.75	81.75	**87.44 ± 0.00**
D⟶W	89.15	99.32	99.32	98.64	98.98	**100.00**	91.80 ± 0.15
W⟶A	62.52	80.58	88.52	75.89	91.34	85.39	**92.38 ± 0.00**
W⟶C	60.37	76.67	83.70	77.11	85.04	80.23	**87.80 ± 0.00**
W⟶D	98.72	**100.00**	**100.00**	**100.00**	99.36	**100.00**	93.50 ± 2.79
Aver	70.37	85.37	87.81	83.33	89.50	84.81	**89.76 ± 0.40**

k-nearest neighbor classifier ($k = 1$) learned from the source domain directly predicts the target samples. Using MMD-penalized kernel PCA, the transfer component analysis (TCA) [24] found common latent features to reduce the marginal distribution discrepancy. The latent characteristics were then used to train a classifier. The joint distribution adaptation (JDA) [23] attempted to construct a robust feature representation by simultaneously narrowing the shift in both the marginal distribution and conditional distribution between domains. A classifier was then trained using these features. By integrating an infinite number of subspaces, the geodesic flow kernel (GFK) [40] learned a robust feature representations. Then, a classifier was learned on these representation. The modified $\mathcal{A}$-distance and sparse filtering (**MASF**) [47] learned common sparse representations by minimizing the modified $\mathcal{A}$-distance between two domains. Then, a classifier was learned from these representations. The joint probability distribution adaptation (JPDA) [48] learned common latent features using modified-MMD (DJP-MMD). Then, a classifier was learned on the latent features. The collaborative representation with curriculum classifier boosting (CRCCB) [49] is aimed at learning directly an adaptive classifier by multistage inference and instance rearranging.

4.5. Results and Analysis. Tables 3–6 show our classification accuracy results as well as the comparable ones for all subtasks ("source"⟶ "target") in the four datasets. The average accuracy of several subtasks is also provided. For statistical verification, we repeat our experiment 20 times on each subtask and give the statistical results. The best and second best results in each subtask are represented by bold and underlined fonts, respectively.

From Tables 3–6, it is concluded that our approach achieves the highest average accuracy consistently on all four datasets. Specifically, on the ImageCLEF-DA dataset (Table 3), our method performs best on almost all subtasks, only slightly worse than CRCCB on two subtasks (1.14%: $I \longrightarrow P$, 0.17%: $P \longrightarrow C$). Our method accomplishes an average accuracy of 88.17%, which is increased by 9.09% (1NN), 6.14% (TCA), 5.38% (JDA), 6.47% (GFK), 5.03% (MASF), 8.07% (JPDA), and 1.49% (CRCCB). On the Office-Caltech10 dataset (Table 4), the average accuracy of ours achieves the best performance. With almost subtasks (9/12), the proposed method wins the first place. Compared with the rest comparison methods, our average accuracy is improved by 19.39% (1NN), 4.39% (TCA), 1.95% (JDA), 6.44% (GFK), 0.26% (MASF), and 4.95% (JPDA).

Table 5: Accuracy (%) on Office31.

Method	1NN	TCA	JDA	GFK	MAFS	JPDA	CRCCB	**CAMLP**
A⟶D	76.91	77.51	78.71	77.51	81.12	77.71	88.55	**88.71 ± 1.13**
A⟶W	76.48	75.72	78.11	73.84	76.98	76.98	84.52	**86.89 ± 0.96**
D⟶A	65.32	67.16	68.62	65.25	62.66	68.19	70.82	**75.04 ± 1.18**
D⟶W	**97.99**	96.86	97.11	96.86	92.08	97.23	96.85	91.62 ± 0.36
W⟶A	62.76	64.68	66.88	63.26	63.61	67.77	68.22	**72.96 ± 1.08**
W⟶D	**99.40**	99.20	99.00	98.59	95.78	99.00	99.39	94.54 ± 1.28
Aver	79.81	80.19	81.40	79.22	78.71	81.15	84.73	**84.96 ± 0.56**

Table 6: Accuracy (%) on Office-Home.

Method	1NN	TCA	JDA	GFK	MAFS	JPDA	CRCCB	**CAMLP**
A⟶C	35.12	35.76	36.06	28.50	29.55	42.15	**47.19**	45.30 ± 0.00
A⟶P	56.79	58.08	58.66	50.48	47.08	56.88	65.71	**71.57 ± 0.71**
A⟶R	64.03	60.98	61.53	55.57	56.14	64.47	71.63	**76.95 ± 0.65**
C⟶A	48.45	42.60	43.22	38.77	36.18	47.30	**63.73**	61.80 ± 0.50
C⟶P	57.85	54.16	56.16	50.89	46.65	60.89	69.52	**75.76 ± 0.51**
C⟶R	59.49	55.34	57.10	51.62	47.49	58.30	70.57	**76.60 ± 0.34**
P⟶A	51.55	47.10	47.51	42.44	38.90	49.24	**64.31**	61.03 ± 0.78
P⟶C	40.73	41.01	41.76	35.14	35.03	44.31	48.01	**48.15 ± 0.50**
P⟶R	68.63	65.80	67.20	61.07	60.78	68.30	78.01	**78.04 ± 0.34**
R⟶A	58.92	55.58	56.12	49.53	49.40	55.58	**70.04**	62.66 ± 1.06
R⟶C	42.73	40.55	43.14	35.33	38.56	47.38	**51.45**	44.61 ± 0.42
R⟶P	73.08	70.06	71.48	66.88	67.52	71.68	80.13	**81.11 ± 1.19**
Aver	54.78	52.25	53.33	47.19	46.11	55.54	65.03	**65.30 ± 0.11**

Table 7: Results of the ablation study. "✓" indicates using the corresponding method, "×" otherwise.

CPLA	MLP	average accuracy (%)			
		ImageCLEF-DA	Office-Caltech	Office31	Office-Home
×	✓	80.28	83.65	79.81	54.80
✓	×	75.79	77.59	70.95	42.29
✓	✓	**88.17 ± 0.20**	**89.76 ± 0.40**	**84.96 ± 0.56**	**65.3 ± 0.11**

In terms of Office31 dataset (Table 5), our method performs the best in four out of six subtasks and ranks first with the average accuracy of 84.96%. Compared with remainder approaches, our average accuracy increases by 5.15% (1NN), 4.77% (TCA), 3.56% (JDA), 5.74% (GFK), 6.26% (MASF), 3.81% (JPDA), and 0.23% (CRCCB). On the Office-Home dataset (Table 6), the average accuracy of ours is 65.30%, which improves by 10.52% (1NN), 13.05% (TCA), 11.97% (JDA), 18.11% (GFK), 19.19% (MASF), 9.76% (JPDA), and 0.27% (CRCCB). In general, the CRCCB and our method performs roughly the same on the two large datasets (Office31, Office-Home), both far better than the other methods. And our method performs best on two relatively small datasets (ImageCLEF-DA, Office-Caltech10).

4.6. Ablation Study. An ablation research is designed to determine how various aspects of our work contributes to the ultimate result. To this goal, we look into various combinations of components such as modified label propagation (MLP) and consistency-aware pseudolabel annotation (CPLA). Specifically, three groups of experiments are conducted. To predict target samples, the first group exclusively uses modified label propagation (MLP) and the initial pseudolabels annotated through a source-domain-trained classifier; the second group directly predicts target samples by CPLA; and the third group combines the MLP and CPLA. To be fair, the three groups use the same 1NN classifier learned only from the source domain. Table 7 shows the average accuracy for the four datasets. It is obvious that

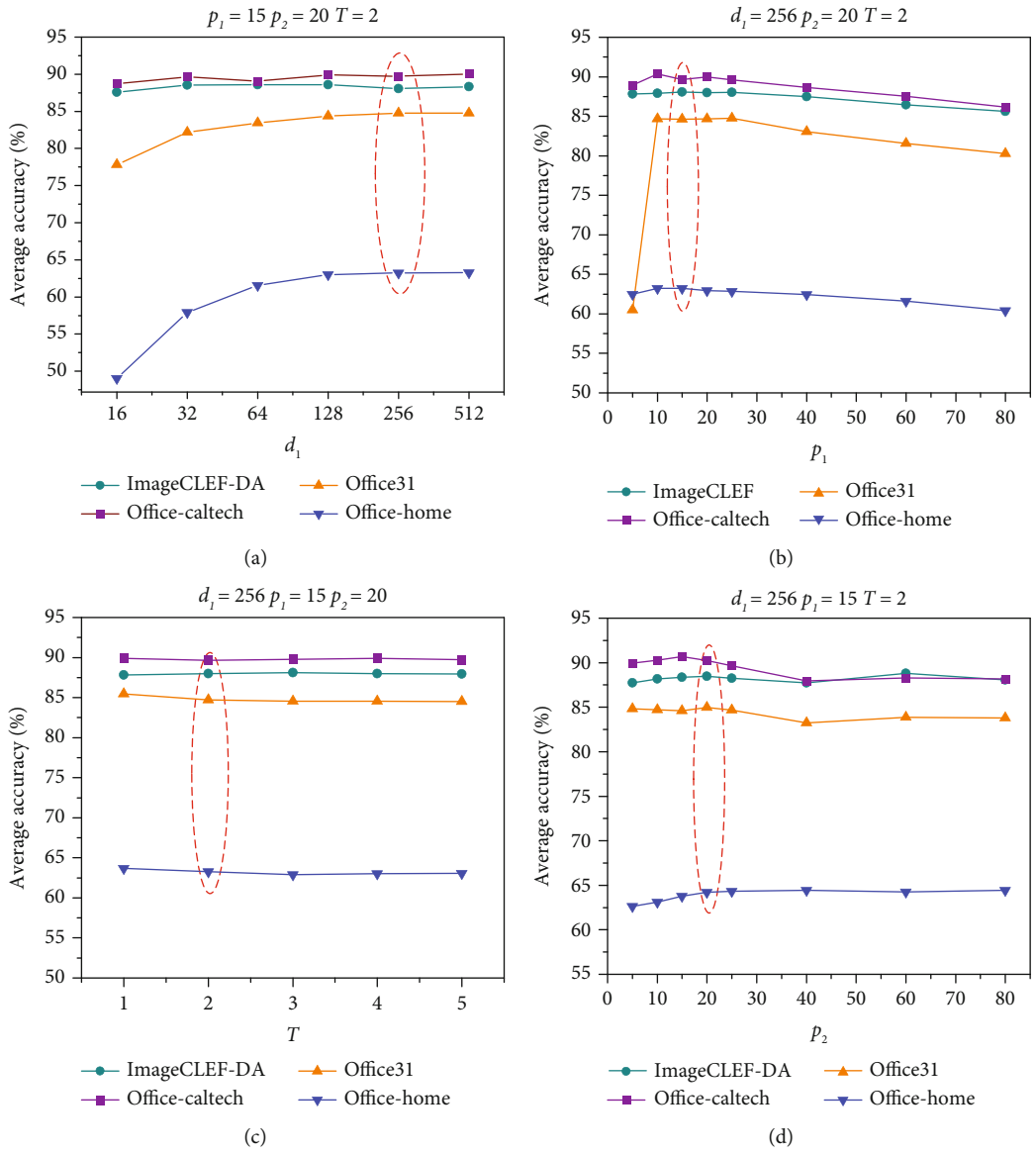

Figure 3: Sensitivity analysis of hyperparameters: (a) PCA dimension d_1, (b) the number of nearest neighbors p_1 in label propagation, (c) the maximum iteration T, and (d) the number of nearest neighbors p_2 in spectral clustering.

MLP and CPLA are ineffective on their own. However, combining them significantly improves their performance.

4.7. Parameter Sensitivity Analysis. Our approach has four hyperparameters: d_1—dimension of PCA space, p_1—number of the nearest neighbors in label propagation, p_2—number of the nearest neighbors in spectral clustering, and T—the maximum iteration. Setting each hyperparameter to a series of different values while fixing others allows us to see how it affects performance. Figure 3 shows the average accuracy across all subtasks, with the parameters used in this paper marked by a red dotted circle. A larger d_1 is obviously necessary for datasets with more categories (Office31, Office-Home), whereas d_1 has little influence on model performance for tasks with few categories (ImageCLEF-DA, Office-Caltech), as shown in Figure 3(a). This is owing to the fact that as the number of categories grows, more features are required. When p_1 is miniscule, the performance drops dramatically on Office31, but less on others datasets. Different p_2 has the greatest influence on the performance of Office-Home (Figure 3(d)). For the four datasets, different T has little effect on the performance, implying that the spectral clustering pseudo labels are nearly the same in each iteration (Figure 3(c)). Our approach swiftly converges after employing the consistent strategy (only one or two iterations).

4.8. Confusion Matrix Visualization. In this part, the performance of CAMLP is compared with that of 1NN (without domain adaptation). Figure 4 illustrates the visualizations of confusion matrix for the two methods on tasks $\mathbf{P} \longrightarrow \mathbf{I}$ and $\mathbf{C} \longrightarrow \mathbf{P}$ from ImageCLEF-DA. Figures 4(a) and 4(b) directly reflect the negative impact brought by domain discrepancy. Clearly, the confusion for most classes is

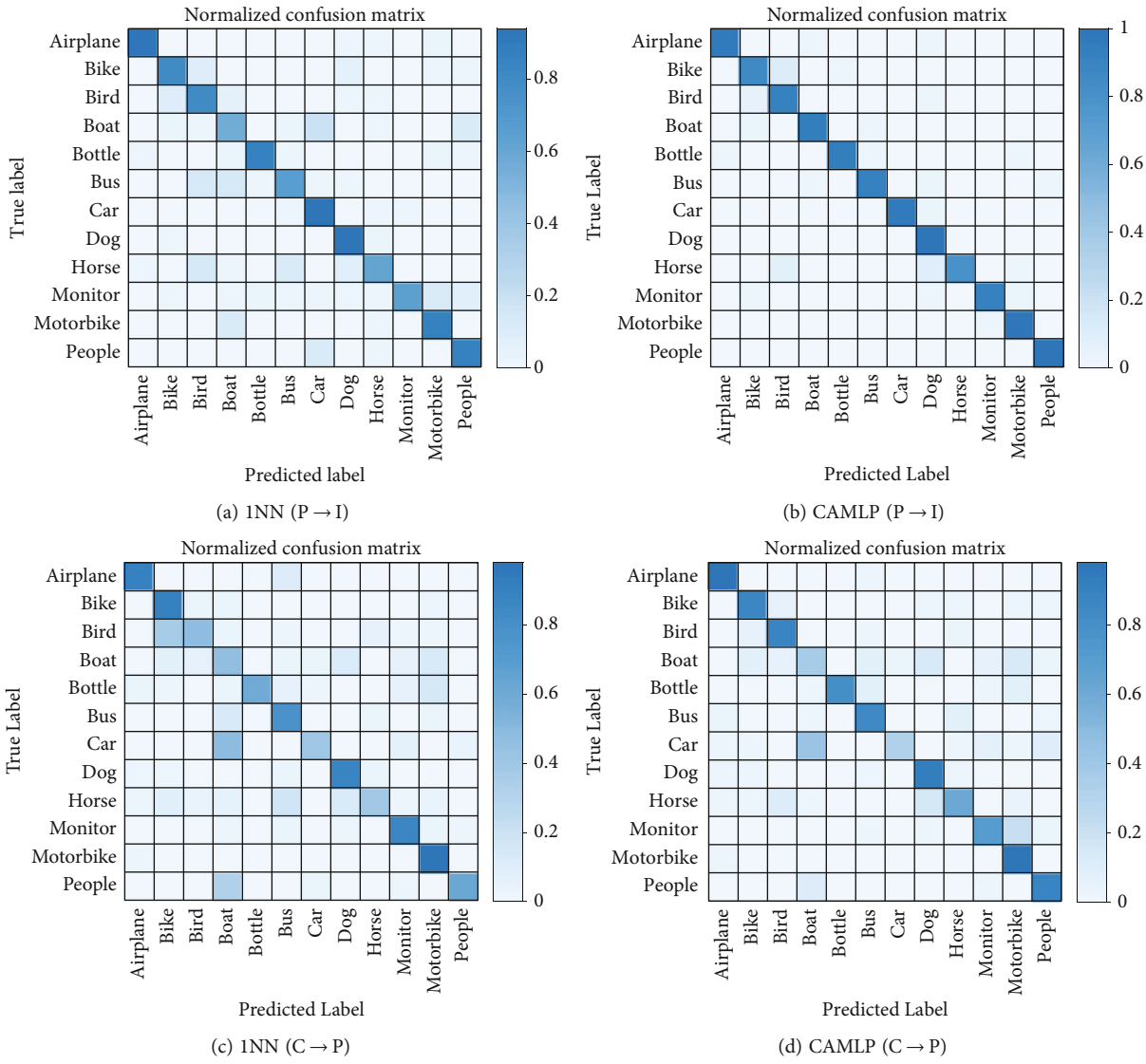

Figure 4: The confusion matrix visualization of 1NN and CAMLP (best viewed in color).

significantly reduced in CAMLP by comparing (a) and (b) and (c) and (d) in Figure 4. For example, the confused boat, bus, and horse are correctly identified by CAMLP in task **P** ⟶ **I**, and the confusion of bike, boat, bottle, and people also is reduced by CAMLP in task **C** ⟶ **P**. Unfortunately, neither CAMLP nor 1NN can handle the problem of misclassifying boat as car on task **C** ⟶ **P**. These observations further reveal that CAMLP has excellent discrimination in the target domain.

5. Discussion

We propose a classifier adaptation method based on modified label propagation. And this method builds different domains in the same graph. We innovatively propose determining relationships between cross-domain samples through pseudolabeling. To alleviate the damage of wrong labels to the model, a consistency-aware pseudolabel annotation method is proposed to improve the accuracy of pseudo labels. Therefore, our method is more suitable for structurally obvious tasks.

According to the experimental results in Section 4, the proposed method achieves better performance in most tasks but has the problem of negative transfer in a few tasks, e.g., $D \longrightarrow W$(Office31), $W \longrightarrow D$(Office31), and $W \longrightarrow D$ (Office-Caltech10). Clearly, the three tasks have one common character: very small domain discrepancy. We attribute this to the fact that our method give more weight to the structural information in the target domain than the knowledge from the source domain. In addition, our method is insensitive to the number of iterations. This maybe be due to the knowledge learned in each iteration is not well inherited. For example, the relationships within the target domain are not updated after each iteration. Addressing this problem will give full play to the advantages of iterative optimization, which is one of the key points of our next work.

6. Conclusion

We propose a classifier adaptation method-based modified label propagation (CAMLP) in this paper. The suggested method consists of two parts: the modified label propagation (MLP) and the consistency-aware pseudolabel annotation (CPLA). MLP builds different-domain samples into a same graph and then transfers the labels from the source domain to the target domain. The CPLA coordinates the source-domain knowledge and the target-domain underlying structure to correct pseudo labels. On the four real-world datasets, sufficient experiments show that the suggested method is superior to state-of-the-art methods. The importance of graph structure information in UDA is demonstrated in this research. For future work, we will investigate the usage of graph neural networks (GNN) in unsupervised domain adaptation, considering that GNN can better extract and utilize structural information among data.

Data Availability

All datasets in the paper are available at an open source repository, https://github.com/jindongwang/transferlearning/tree/master/data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62076204 and 62106134), the National Natural Science Foundation of Shaanxi Province (Grant Nos. 2018JQ6003 and 2018JQ6030), and the Postdoctoral Science Foundation of China under Grant 2021M700337.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: a survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[3] M. Wang and W. Deng, "Deep visual domain adaptation: a survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[4] E. Ahn, A. Kumar, M. Fulham, D. Feng, and J. Kim, "Unsupervised domain adaptation to classify medical images using zero-bias convolutional auto-encoders and context-based feature augmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 7, pp. 2385–2394, 2020.

[5] Y. Zhang, Y. Wei, W. Qingyao et al., "Collaborative unsupervised domain adaptation for medical image diagnosis," *IEEE Transactions on Image Processing*, vol. 29, pp. 7834–7844, 2020.

[6] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, Eds., "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 994–1003, Salt Lake City, Utah, 2018.

[7] L. T. Triess, M. Dreissig, C. B. Rist, and J. Marius Zöllner, "A survey on deep domain adaptation for LiDAR perception," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pp. 350–357, Nagoya, Japan, 2021.

[8] Y. Wang, J. Peng, H. Wang, and M. Wang, "Progressive learning with multi-scale attention network for cross-domain vehicle re-identification," *Science China Information Sciences*, vol. 65, no. 6, pp. 1–15, 2022.

[9] D. Hussain, M. Ismail, I. Hussain, R. Alroobaea, S. Hussain, and S. S. Ullah, "Face mask detection using deep convolutional neural network and MobileNetV2-based transfer learning," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1536318, 10 pages, 2022.

[10] X. Ma and X. Li, "Dynamic gesture contour feature extraction method using residual network transfer learning," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 1503325, 11 pages, 2021.

[11] J. He, J. Xu, S. Chen, and J. Liu, "Multi-source domain adaptation with collaborative learning for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11008–11017, June 2021.

[12] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 1215–1224, June 2021.

[13] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation: a review," *Technologies (Basel)*, vol. 8, no. 2, p. 35, 2020.

[14] Y. Zou, Y. Zhiding, B. V. K. Vijaya Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the european conference on computer vision (ECCV)*, pp. 289–305, Munich Germany, September 2018.

[15] F. Yang, Y. Wei, G. Wang, Y. Zhou, H. Shi, and S. Thomas, "Self-similarity grouping: a simple unsupervised cross domain adaptation approach for person re-identification," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp. 6112–6121, Seoul Korea, October 2019.

[16] D. Mekhazni, A. Bhuiyan, G. Ekladious, and E. Granger, "Unsupervised domain adaptation in the dissimilarity space for person re-identification," in *European Conference on Computer Vision (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., pp. 159–174, Springer International Publishing, Cham, 2020.

[17] H. Tang, Y. Zhao, and L. Hongtao, "Unsupervised person re-identification with iterative self-supervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops*, Long Beach, CA, June 2019.

[18] M. Abdelwahab and C. Busso, "Supervised domain adaptation for emotion recognition from speech," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5058–5062, South Brisbane, Queensland, Australia,, 2015.

[19] B. Li, Y. Wang, S. Zhang et al., "Learning invariant representations and risks for semi-supervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 1104–1113, June 2021.

[20] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in

Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp. 8050–8058, Seoul, Korea, October 2019.

[21] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 769–776, Sydney, NSW, Australia, 2013.

[22] A. Singh, "CLDA: contrastive learning for semi-supervised domain adaptation," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, Eds., vol. 34, pp. 5089–5101, Curran Associates, Inc., 2021.

[23] M. Long, J. Wang, G. Ding, J. Sun, and S. Y. Philip, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2200–2207, Sydney, NSW, Australia, December 2013.

[24] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[25] J. Wang, W. Feng, Y. Chen, Y. Han, M. Huang, and S. Y. Philip, "Visual domain adaptation with manifold embedded distribution alignment," in *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 402–410, New York, NY, USA, 2018.

[26] Z. Ding, S. Li, M. Shao, and F. Yun, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision*, pp. 37–52, Munich, Germany, September 2018.

[27] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, "Transfer learning with graph co-regularization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1805–1818, 2014.

[28] Q. Wang and T. P. Breckon, "Unsupervised domain adaptation via structured prediction based selective pseudo-labeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6243–6250, New York, NY, USA, 2020.

[29] T. M. H. Hsu, W. Y. Chen, C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, Eds., "Unsupervised domain iadaptation with imbalanced cross-domain data," in *Proceedings IEEE International Conference on Computer Vision*, pp. 4121–4129, Santiago, Chile, December 2015.

[30] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, Phoenix, Arizona USA, March 2016.

[31] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3934–3941, New Orleans, LA, 2018.

[32] Q. Wang, B. Penghui, and T. P. Breckon, "Unifying unsupervised domain adaptation and zero-shot visual recognition," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8, Budapest, Hungary, July 2019.

[33] W. Zhang, W. Ouyang, W. Li, and X. Dong, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3801–3809, Salt Lake City, UT, 2018.

[34] J. Liang, R. He, Z. Sun, and T. Tan, "Exploring uncertainty in pseudo-label guided unsupervised domain adaptation," *Pattern Recognition*, vol. 96, p. 106996, 2019.

[35] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 9, pp. 1010–1015, Pasadena, California, 2009.

[36] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5552–5562, 2016.

[37] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 321–328, Vancouver, Canadayear, 2003.

[38] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[39] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[40] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2066–2073, Providence, Rhode Island, 2012.

[41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision (ECCV)*, pp. 213–226, Berlin Heidelberg, 2010.

[42] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5018–5027, Honolulu, Hawaii, USA, July 2017.

[43] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, Eds., "DeCAF: a deep convolutional activation feature for generic visual recognition," in *Proceedings of the International Conference on Machine Learning*, pp. 647–655, Beijing, China, 2014.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, Nevada, 2016.

[45] C. Han, Y. Deyun Zhou, X.,. Y. Lei, J. Shi, and M. Gong, Eds., "Discrepancy-aware collaborative representation for unsupervised domain adaptation," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–6, Glasgow, United Kingdom, 2020.

[46] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[47] Y. Chao Han, L.,. Y. Xie, D. Zhou, and M. Gong, "Visual domain adaptation based on modified A−distance and sparse filtering," *Pattern Recognition*, vol. 104, p. 107254, 2020.

[48] W. Zhang and W. Dongrui, "Discriminative joint probability maximum mean discrepancy (DJP-MMD) for domain adaptation," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8, Glasgow, UK, 2020.

[49] C. Han, Y. Deyun Zhou, M. G. Xie, Y. Lei, and J. Shi, "Collaborative representation with curriculum classifier boosting for unsupervised domain adaptation," *Pattern Recognition*, vol. 113, p. 107802, 2021.