Probability-Based Graph Embedding Cross-Domain and Class Discriminative Feature Learning for Domain Adaptation

Wenxu Wang, Zhencai Shen, Daoliang Li, Ping Zhong, and Yingyi Chen

***Abstract*— Feature-based domain adaptation methods project samples from different domains into the same feature space and try to align the distribution of two domains to learn an effective transferable model. The vital problem is how to find a proper way to reduce the domain shift and improve the discriminability of features. To address the above issues, we propose a unified Probability-based Graph embedding Cross-domain and class Discriminative feature learning framework for unsupervised domain adaptation (PGCD). Specifically, we propose novel graph embedding structures to be the class discriminative transfer feature learning item and cross-domain alignment item, which can make the same-category samples compact in each domain, and fully align the local and global geometric structure across domains. Besides, two theoretical analyses are given to prove the interpretability of the proposed graph structures, which can further describe the relationships between samples to samples in single-domain and cross-domain transfer feature learning scenarios. Moreover, we adopt novel weight strategies via probability information to generate robust centroids in each proposed item to enhance the accuracy of transfer feature learning and reduce the error accumulation. Compared with the advanced approaches by comprehensive experiments, the promising performance on the benchmark datasets verify the effectiveness of the proposed model.**

***Index Terms*— Graph embedding, class discriminative feature learning, cross-domain alignment, probability information, unsupervised domain adaptation.**

I. Introduction

IN THE data-driven era, abundant datasets in practical applications are difficult to meet the independent and identically distributed conditions [1] which are assumptions of supervised learning paradigms due to the different collection environments. Under this circumstance, the model learned on the training data (source domain) will fail to classify the testing data (target domain) with the different distribution. Hence, how to utilize a well-labeled source domain to provide reliable knowledge for the unlabeled target domain is a critical issue in designing the effective models. Unsupervised domain adaptation (UDA) in transfer learning can provide solutions for the scenarios [2].

UDA has been successfully applied in image classification [3], image segmentation [4], image retrieval [5], object detection [6], and many other areas. Recently, the feature-based methods of UDA have drawn great attention owing to their great performances. Among them, one of the most common methods is to minimize the difference between features of different domains in the projection space through the suitable distance metric methods, such as the maximum mean discrepancy (MMD) [7], [8]. However, the compactness of each domain may be sacrificed during the process of minimizing MMD to make the domain distribution close [9], which may cause the feature distortion and damage the intrinsic class-wise structures transferred from source to target [10]. In order to keep the compactness and improve the class discriminativeness of each domain, many existing methods try to maximize the inter-class dispersion or minimize the intra-class scatter, or both [9], [10], [11], [12], and the commonly used method for minimizing the intra-class scatter is achieved through the minimization of the distance between samples and class center. However, due to the inaccuracy of the pseudo labels, the errors are inevitably introduced in the resulting class centers in target domain, which will cause the discriminativeness deterioration of the learned transfer features.

Inspired by the conditional posterior probability in [13], we utilize the probabilities based on robust soft labels to construct the novel weighted centroids for target samples. Specifically, besides constructing target centroids for each category by the confidence weights like [13], we further weight the different centroids via the target soft pseudo label to obtain a specific weighted centroid for each target sample. The weighted centroids are adopted to establish a new target class discriminative feature learning item, which can reduce the error accumulation of pseudo labels. The distances between each target sample and its corresponding weighted centroid are minimized to make the intra-class compactness, so as to enhance the discriminativeness of the learned transfer features.

Manuscript received 24 September 2021; revised 21 August 2022; accepted 19 November 2022. Date of publication 7 December 2022; date of current version 16 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62076244, in part by the Chinese Universities Scientific Fund under Grant 2022TC109, in part by the Double First-Class Project of China Agricultural University (2022), and in part by the Double First-Class International Cooperation Project of China Agricultural University under Grant 10020799. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kui Jia. *(Corresponding authors: Ping Zhong; Yingyi Chen.)*

Wenxu Wang is with the College of Information and Electrical Engineering, National Innovation Center for Digital Fishery, China Agricultural University, Beijing 100083, China (e-mail: wangwenxu93@126.com).

Zhencai Shen and Ping Zhong are with the College of Science, National Innovation Center for Digital Fishery, China Agricultural University, Beijing 100083, China (e-mail: zhencai688@cau.edu.cn; zping@cau.edu.cn).

Daoliang Li and Yingyi Chen are with the College of Information and Electrical Engineering, National Innovation Center for Digital Fishery, Beijing Engineering and Technology Research Center for Internet of Things in Agriculture, Precision Agricultural Technology Integration Research Base (Fishery), Ministry of Agriculture and Rural Affairs, and the Key Laboratory of Agricultural Information Acquisition, Ministry of Agriculture, China Agricultural University, Beijing 100083, China (e-mail: dliangl@cau.edu.cn; chenyingyi@cau.edu.cn).

Digital Object Identifier 10.1109/TIP.2022.3226405

Besides learning the discriminative transfer features, aligning the geometric structure of two domains to reduce the domain shift is another key issue. To achieve this goal, the class center alignment including the MMD-based joint distribution alignment and other similar strategies are generally adopted [14], [15], [16], [17]. However, focusing only on the class center alignment can not eliminate the cross-domain bias of samples, which will result in the unsatisfactory alignment of the same-category samples in two domains.

To solve this problem, we propose a novel cross-domain alignment item that can align the global and local geometry structures of the two domains from the perspective of samples. On one hand, the global cross-domain alignment item makes each source sample close to its same-category target centroid. On the other hand, we construct a specific weighted source centroid for each target sample based on the confidence weight and the soft-label category aggregation of the target sample, then impose each target sample close to the weighted source centroids. In this way, the global spatial relationship of two domains can be preserved in the projection space. Moreover, each source sample is imposed to be close to the same-category nearest neighbors in the target domain through the proposed local cross-domain alignment item to preserve the local spatial relationship in the low-dimensional projection space.

By noticing that the proposed feature learning items and alignment items can generally be expressed as the distances from the samples to the weighted centroids, we further prove that by giving the appropriate adjacency matrices, these items can be transformed into graph embedding structures with fully interpretable to represent the sample-to-sample relationship. Based on this, we propose a unified Probability-based Graph embedding Cross-domain and class Discriminative feature learning framework (PGCD) as shown in Fig. 1. In the framework, the geometric structure of the two domains can be fully aligned, and the discriminability of each domains can be enhanced. Thereby, the classifier trained on source samples can also achieve the good performance on the target domain.

Above all, the main contributions of this paper are as follows:

- We propose the graph embedding domain adaptation framework PGCD. It can efficiently align the distribution of the two domains from the sample viewpoint and learn discriminative features to avoid distortion of projected features.
- We prove that the target class discriminative feature learning item and the global cross-domain alignment item can be expressed as graph structures. By constructing interpretable adjacency matrices, we can have a more comprehensive understanding of the relationship between samples in cross-domain alignment and class discriminative learning scenarios.
- We adopt novel weight strategies via probability information to generate robust centroids in target class discriminative item and global cross-domain item to enhance the accuracy of transfer feature learning and reduce the error accumulation.

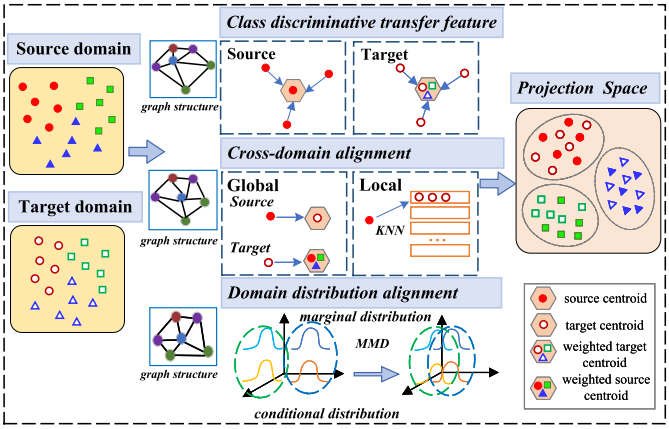


Fig. 1. Schematic of the proposed PGCD. (PGCD contains the class discriminative feature learning item, the cross-domian alignment item, and the domain distribution alignment item. Each item is embedded in PGCD as a graph embedding structure.)

- We conducted extensive experiments on the popular benchmark datasets, and the results demonstrate the superiority of the proposed method PGCD.

The rest of this paper is organized as follows: Section 2 reviews the related work; in Section 3, the proposed PGCD is introduced in detail, including the theorems and computational complexity of the algorithm; Section 4 analyzes the empirical studies on benchmark datasets; Section 5 concludes the paper.

II. Related Work

A. Shallow UDA Methods

Shallow UDA methods can be grouped into three main categories [2]: the instance-based adaptation [18], [19], the model-based adaptation [20], [21], and the feature-based adaptation. The extensively applied feature-based adaptation method can be further divided into subspace-centric methods and data-centric methods. Subspace-centric methods directly align two domains rather than discovering a new common feature space [22], [23]. Data-centric methods project features into a projection space and reduce the gap between data distributions of two domains to learn the domain-invariant features [2]. The proposed method PGCD falls into the category of the data-centric feature-based adaptation.

In data-centric methods, the classic transfer component analysis (TCA) [7] aligned the marginal distribution of the two domains, and on this basis, joint distribution adaptation (JDA) [8] further aligned the conditional distribution. Zhang et al. [24] designed two projections to achieve the geometrical alignment. Liang et al. [13] proposed a probability-based joint distribution alignment item and utilized a triplet-wise strategy to align the cross-domain local structure. Liu et al. [25] proposed a novel homologous component analysis (HCA) method which can find two totally different but homologous transformations to align distributions. Tian et al. [16] regarded the samples in the same-category cluster in target domain as a whole, and utilized the adaptive local manifold self-learning to capture the connectivity. Jing et al. [17] aligned the first-order statistics

and second-order statistics, and classified the aligned feature by optimizing the structural risk in the reproducing kernel Hilbert space (RKHS). Wang et al. [26] adopted dynamic distribution alignment to explore the relative importance between marginal and conditional distributions in Grassmann manifold.

For learning the discriminative features, Li et al. [11] implemented a feature learning method to make intra-class compactness and inter-class dispersion, further, they refined the target labels by the label consistency strategy, which made the transfer feature learning and label refinement mutually beneficial [10]. Liang et al. [3] proposed a relaxed domain-irrelevant clustering-promoting item to find the optimal projection, which can bridge the semantic gap between domains. Lu et al. [27] used the mean value of each class to learn the class-specific linear projections. Yang et al. [12] incorporated the minimum intra-class scatter into the process of domain adaptation to increase the discriminativeness of two domains, further, they imposed a row-sparsity constraint on the transformation matrix to improve the classification performance [14]. Xie et al. [15] proposed a joint metric and feature representation learning method (JMFL) for UDA, which considers the feature representation learning, the metric learning, and the discriminative learning. Chen et al. [9] deeply analyzed MMD from the perspective of graph structure, and proposed an MMD-based graph embedding framework that contains the classic dimensionality reduction PCA, LDA, MFA and metric learning LMNN to learn discriminative features. In order to enhance the robustness of UDA, Ding et al. [28] investigated the robust transfer metric learning, and Wang et al. [29] deeply investigated the robust property of kernel mean p-power error loss.

Among these related studies, most of them are devoted to the alignment of two domains from the perspective of the class center, and the same target center is shared for all target samples in each class due to the hard pseudo labels. The above center-wise alignment strategy cannot sufficiently eliminate the cross-domain bias of sample level. In order to align two domains more comprehensively, we propose the cross-domain alignment item from sample to sample instead of the current class center strategy. We utilize the nearest neighbor and weighted centroid strategies to align the local and global geometry structures of the two domains and to find specific alignment targets for each sample, which can achieve sufficient domain alignment. Besides, we adopt the soft labels to reduce the error accumulation when calculating the centroids, so as to learn the more discriminative transfer features.

B. Deep UDA Methods

In recent years, the deep UDA methods based on the metric learning strategy [30], [31], [32], [33], the adversarial learning strategy [34], [35], [36], [37], [38], [39], [40], [41], and other specific alignment modules [42], [43], [44], [45], [46] have performed well in various fields and attracted the increasing attention of researchers.

The metric-based UDA methods achieve the good performance by reducing the difference between domains through MMD. For instance, Tzeng et al. [30] added an MMD-based adaptive layer to adapt the data of both domains. Long et al. [31] adopted the multi-kernel MMD metric to align two domain, further, they proposed a joint maximum mean discrepancy (JMMD) metric to align the joint distributions [32]. Hu et al. [33] utilized hierarchical domain alignments to learn domain-invariant features.

The adversarial-based UDA methods use adversarial strategy to learn deep domain-invariant features and align the distribution. For instance, Ganin et al. [34] pioneered the idea to add the adversarial mechanism to domain adaption. Long et al. [35] proposed a conditional domain adversarial network (CDAN) to reduce domain shift. Liu et al. [36] generated the transferable samples to fill the domain gap and trained the deep classifiers on them adversarially. Chen et al. [37] proposed an attention module in the discriminator to discriminate the transferable regions among the samples of both domains. Yang et al. [38] proposed a geometry-aware dual-stream network to learn the geometry-aligned representations to handle the problem of inconsistent geometries. Ma et al. [39] improved the discriminability and the transferability of feature representations by minimizing the entropy of the independent domain and maximizing the entropy across domains. Cui et al. [40] proposed a gradually vanishing bridge mechanism for adversarial learning, which can reduce the impact of domain-specific features. Gu et al. [41] proposed an adversarial UDA approach in spherical feature space with robust pseudo labels.

Some researchers have explored new deep domain adaptation modules combining different techniques. Tang et al. [42] proposed a novel structurally regularized deep clustering method, and constrained the clustering solutions by structural source regularization. Na et al. [43] trained the source-dominant model and the target-dominant model with complementary features to achieve knowledge transfer. Yue et al. [44] proposed the transporting causal mechanisms to learn the domain-invariant features. Li et al. [45] proposed a domain conditioned adaptation network that explored the low-level domain-dependent knowledge and aligned high-level domain distributions. Xu et al. [46] proposed a weight-sharing triple-branch transformer framework with a two-way pseudo-label strategy to achieve domain adaptation.

Compared with the deep UDA methods, the proposed PGCD is a lightweight method without using deep neural network structures. We utilize the interpretable graph structures to build cross-domain alignment and class discriminative feature learning items, which can benefit the understanding of the relationship between samples in UDA scenarios. As documented by many studies [10], [14], [16] shallow UDA paradigms equipped with the general deep features have achieved comparable performance to the deep ones, which demonstrates that deep feature learning and domain alignment can be in separate ways. In the present study, we learn the class discriminative features and align the two domains based on the deep features, which indeed produced the desired results.

III. The Proposed Method PGCD

A. Soft Labels and Probability Information

In UDA paradigms, there are two domains: a fully labeled source domain $\mathcal{D}_s = \{(\mathbf{x}_{si}, \mathbf{y}_{si})\}_{i=1}^{n_s} = \{\mathbf{X}_s, \mathbf{y}_s\}$ and an

unlabeled target domain $\mathcal{D}_t = \{\mathbf{x}_{tj}\}_{j=1}^{n_t} = \mathbf{X}_t$. The source domain data matrix $\mathbf{X}_s \in \mathbb{R}^{d\times n_s}$ contains n_s samples with d-dimension and the target domain data matrix $\mathbf{X}_t \in \mathbb{R}^{d\times n_t}$ contains n_t samples with d-dimension. The total data matrix $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t] \in \mathbb{R}^{d\times(n_s+n_t)}$ contains $n = n_s + n_t$ samples. Both domains under the assumptions that the feature space $\mathcal{X}_s = \mathcal{X}_t$, the label space $\mathcal{Y}_s = \mathcal{Y}_t$, $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$ and $P_s(\mathbf{y}_s \mid \mathbf{x}_s) \neq P_t(\mathbf{y}_t \mid \mathbf{x}_t)$, which means that although two domains have the same feature space and categories, their marginal distribution and the conditional distribution are different.

Since the target samples are unlabeled, the pseudo labels are commonly used as their categories. The usual practice is to use the hard pseudo labels which are predicted by the classifier trained on the projected source domain as the labels of target samples. However, the inaccuracy of the predicted labels will cause errors of the transfer feature learning, especially when it occurs at the beginning of iteration, the error accumulation with iterations will have a great impact on the final classification result. To alleviate the error accumulation, PGCD adopts the K-nearest neighbor classifier in the projection space to generate soft pseudo labels for the target samples. First, we define the c-th class soft label value of a target sample:

$$\mathrm{k}(\mathbf{x}_{ti}, c) = p^c/p \tag{1}$$

where p is the number of nearest neighbors of the target sample $\mathbf{x}_{ti}$ in the projected source domain; p^c is the number of the c-th class samples in the nearest neighbor set. As a target sample could belong to several categories, we denote the category aggregation of $\mathbf{x}_{ti}$ as Ω_{ti}:

$$\Omega_{ti} = \{\ c \mid \mathrm{k}(\mathbf{x}_{ti}, c) > 0,\ \ c = 1, \ldots, C\ \} \tag{2}$$

If the soft label value $\mathrm{k}(\mathbf{x}_{ti}, c^*)$ is greater than 0, then class c^* belongs to the category aggregation Ω_{ti}. Conversely, if $\mathrm{k}(\mathbf{x}_{ti}, c^*)$ equals to 0, then c^* is not in the Ω_{ti}, and $\mathbf{x}_{ti}$ does not belong to the class c^*. Then the confidence weight of $\mathbf{x}_{ti}$ of the c-th class is defined as

$$\mathrm{g}(\mathbf{x}_{ti}, c) = \mathrm{k}(\mathbf{x}_{ti}, c) \Big/ \sum_{x_{tj}\in\mathbf{X}_t^c} \mathrm{k}(\mathbf{x}_{tj}, c) \tag{3}$$

If $\mathbf{x}_{ti}$ does not belong to the class c^*, $\mathrm{g}(\mathbf{x}_{ti}, c^*)$ equals to 0. The confidence weight expresses the importance degree of $\mathbf{x}_{ti}$ in class c.

B. Class Discriminative Transfer Feature Learning With Probability Information

In order to learn more discriminative transfer feature, PGCD proposes the graph-based class discriminative items for both domains by minimizing the distance between the sample to its centroid, which can make intra-class compact to avoid the feature distortion. Especially in the target domain, confidence weight is used to contruct the specific weighted centroid for each target sample, which can enhance the accuracy of transfer feature learning.

1) Source Class Discriminative Transfer Feature Learning: For learning the class discriminative features in the source domain, the distances between samples and the intra-class centroid of the same class are minimized as follows:

$$\min_{\mathbf{A}} \sum_{c=1}^{C} \sum_{\mathbf{x}_{si}\in\mathcal{D}_s^c} \left\| \mathbf{A}^\top \mathbf{x}_{si} - \mathbf{m}_s(c) \right\|^2 \tag{4}$$

where $\mathbf{m}_s(c)$ is the source centroid of the c-th class, and $\mathbf{m}_s(c) = \frac{1}{n_s^c} \sum_{\mathbf{x}_{sj}\in\mathcal{D}_s^c} \mathbf{A}^\top \mathbf{x}_{sj}$.

Proposition 1: For the c-th class samples of the source domain, the sum of the distances between samples and the intra-class centroid can be expressed as a graph embedding structure:

$$\sum_{\boldsymbol{x}_{si}\in\mathcal{D}_s^c} \left\| \boldsymbol{A}^\top \boldsymbol{x}_{si} - \boldsymbol{m}_s(c) \right\|^2 = \sum_{\boldsymbol{x}_{si},\boldsymbol{x}_{sj}\in\mathcal{D}_s^c} \left\| \boldsymbol{A}^\top \boldsymbol{x}_{si} - \boldsymbol{A}^\top \boldsymbol{x}_{sj} \right\|^2 W_{ij}^{I_1^c} \tag{5}$$

where $W_{ij}^{I_1^c} = \begin{cases} \frac{1}{2n_s^c},\ \boldsymbol{x}_{si} \in \mathcal{D}_s^c, \boldsymbol{x}_{sj} \in \mathcal{D}_s^c \\ 0, \quad \textit{otherwise.} \end{cases}$.

Proposition 1 is obvious according to the theorem of cluster [47]. In addition, we have

$$\sum_{\mathbf{x}_{si}\in\mathcal{D}_s} \sum_{\mathbf{x}_{sj}\in\mathcal{D}_s} \left\| \mathbf{A}^\top \mathbf{x}_{si} - \mathbf{A}^\top \mathbf{x}_{sj} \right\|^2 W_{ij}^{I_1} = \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}_s \mathbf{L}^{I_1} \mathbf{X}_s \mathbf{A} \right) \tag{6}$$

where $W_{ij}^{I_1} = \begin{cases} \frac{1}{2n_s^c},\ x_i \in \mathcal{D}_s^c,\ x_j \in \mathcal{D}_s^c,\ c = 1, 2, \ldots, C \\ 0, \quad \text{otherwise.} \end{cases}$, $\mathbf{L}^{I_1} \in \mathbb{R}^{n_s\times n_s}$ is a Laplacian matrix, $\mathbf{L}^{I_1} = \mathbf{D}^{I_1} - \mathbf{W}^{I_1}$, and $\mathbf{D}^{I_1}$ is a diagonal matrix with $D_{ii}^{I_1} = \sum_j W_{ij}^{I_1}$. Formula Eq.(4) can be transformed as

$$\min_{\mathbf{A}} \sum_{c=1}^{C} \sum_{\mathbf{x}_{si}\in\mathcal{D}_s^c} \left\| \mathbf{A}^\top \mathbf{x}_{si} - \mathbf{m}_s(c) \right\|^2 = \min_{\mathbf{A}} \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}_s \mathbf{L}^{I_1} \mathbf{X}_s \mathbf{A} \right) \tag{7}$$

2) Target Class Discriminative Transfer Feature Learning: In order to improve the discriminativeness of target transfer features, we propose a specific weighted centroid strategy to make intra-class of target domain compact.

In target domain, the c-th class centroid [13] is defined as

$$\mathbf{m}_t(c) = \sum_{\mathbf{x}_{tj}\in\mathcal{D}_t^c} \mathrm{g}(\mathbf{x}_{tj}, c) \mathbf{A}^\top \mathbf{x}_{tj} \tag{8}$$

where $\mathrm{g}(\mathbf{x}_{tj}, c)$ is the confidence weight defined in Eq.(3). Since the category aggregation Ω_{ti} (defined in Eq.(2)) and the soft label value of each target sample could be different, each target sample $\mathbf{x}_{ti}$ has its unique weighted centroid $\mathbf{m}_t(\mathbf{x}_{ti}, \Omega_{ti})$ which we define as follows:

$$\mathbf{m}_t(\mathbf{x}_{ti}, \Omega_{ti}) = \sum_{c\in\Omega_{ti}} \mathrm{k}(\mathbf{x}_{ti}, c)\, \mathbf{m}_t(c) \tag{9}$$

where $\mathrm{k}(\mathbf{x}_{ti}, c)$, $\mathbf{m}_t(c)$ are defined by Eq.(1) and Eq.(8), respectively. Denote

$$\Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}) = \sum_{c\in\Omega_{ti}} \mathrm{k}(\mathbf{x}_{ti}, c)\, \mathrm{g}(\mathbf{x}_{tj}, c) \tag{10}$$

Noticing that if $\mathbf{x}_{tj} \notin \mathcal{D}_t^c$, we have $\mathrm{g}(\mathbf{x}_{tj}, c) = 0$. So Eq.(10) can be transformed as $\Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}) = \sum_{c \in \Omega^*} \mathrm{k}(\mathbf{x}_{ti}, c)\, \mathrm{g}(\mathbf{x}_{tj}, c)$, where Ω^* is the common category aggregation of $\mathbf{x}_{ti}$ and $\mathbf{x}_{tj}$. Based on this and the definition of $\mathrm{g}(.,.)$, we have $\Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}) = \Psi(\mathbf{x}_{tj}, \mathbf{x}_{ti}, \Omega_{tj})$. And we have

$$\mathbf{m}_t(\mathbf{x}_{ti}, \Omega_{ti}) = \sum_{\mathbf{x}_{tj} \in \mathcal{D}_t} \Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}) \mathbf{A}^\top \mathbf{x}_{tj} \tag{11}$$

To learn the class discriminative features in target domain, the distances from each sample to its corresponding multi-class centroid are minimized as follows:

$$\min_{\mathbf{A}} \sum_{\mathbf{x}_{ti} \in \mathcal{D}_t} \left\| \mathbf{A}^\top \mathbf{x}_{ti} - \mathbf{m}_t(\mathbf{x}_{ti}, \Omega_{ti}) \right\|^2 \tag{12}$$

Theorem 1: In target domain, the sum of the distances between samples and their corresponding specific weighted centroids can be expressed as a graph embedding structure:

$$\sum_{i=1}^{n_t} \left\| \boldsymbol{A}^\top \boldsymbol{x}_{ti} - \boldsymbol{m}_t(\boldsymbol{x}_{ti}, \Omega_{ti}) \right\|^2 = \frac{1}{2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \left\| \boldsymbol{A}^\top \boldsymbol{x}_{ti} - \boldsymbol{A}^\top \boldsymbol{x}_{tj} \right\|^2 W_{ij}^{I_2} \tag{13}$$

where

$$W_{ij}^{I_2} = \begin{cases} -\sum_{q=1}^{n_t} \Psi(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tq}, \Omega_{ti}) \Psi(\boldsymbol{x}_{tj}, \boldsymbol{x}_{tq}, \Omega_{tj}) \\ \qquad + 2\Psi(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj}, \Omega_{ti}), & i \neq j \\ 0, & i = j \end{cases} \tag{14}$$

where $\Psi(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj}, \Omega_{ti})$ is defined in Eq.(10).

Proof: Given $W_{ij}^{I_2}$, since $\Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}) = \Psi(\mathbf{x}_{tj}, \mathbf{x}_{ti}, \Omega_{tj})$, the adjacency matrix $\mathbf{W}^{I_2}$ is a symmetric matrix. By denoting $\mathbf{z}_{ti} = \mathbf{A}^\top \mathbf{x}_{ti}$, $\mathbf{z}_{tj} = \mathbf{A}^\top \mathbf{x}_{tj}$, we have

$$\begin{aligned} &\sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \left\| \mathbf{A}^\top \mathbf{x}_{ti} - \mathbf{A}^\top \mathbf{x}_{tj} \right\|^2 W_{ij}^{I_2} \\ &= \sum_{i=1}^{n_t} \left(\sum_{j=1}^{n_t} W_{ij}^{I_2} \right) \mathbf{z}_{ti}^\top \mathbf{z}_{ti} + \sum_{j=1}^{n_t} \left(\sum_{i=1}^{n_t} W_{ij}^{I_2} \right) \mathbf{z}_{tj}^\top \mathbf{z}_{tj} \\ &\quad - 2 \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \mathbf{z}_{ti}^\top \mathbf{z}_{tj} W_{ij}^{I_2} \\ &= 2 \sum_{i=1}^{n_t} D_i^{I_2} \mathbf{z}_{ti}^\top \mathbf{z}_{ti} - 2 \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \mathbf{z}_{ti}^\top \mathbf{z}_{tj} W_{ij}^{I_2} \\ &= 2 \,\mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}_t (\mathbf{D}^{I_2} - \mathbf{W}^{I_2}) \mathbf{X}_t^\top \mathbf{A} \right) \\ &= 2 \,\mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}_t \mathbf{L}^{I_2} \mathbf{X}_t{}^\top \mathbf{A} \right) \end{aligned} \tag{15}$$

where $\mathbf{L}^{I_2} \in \mathbb{R}^{n_t \times n_t}$ is a Laplacian matrix, $\mathbf{L}^{I_2} = \mathbf{D}^{I_2} - \mathbf{W}^{I_2}$, $\mathbf{D}^{I_2}$ is a diagonal matrix with $D_{ii}^{I_2} = \sum_j W_{ij}^{I_2}$, and

$$(\mathbf{L}^{I_2})_{ij} = \begin{cases} \sum_{q=1}^{n_t} \Psi(\mathbf{x}_{ti}, \mathbf{x}_{tq}, \Omega_{ti}) \Psi(\mathbf{x}_{tj}, \mathbf{x}_{tq}, \Omega_{tj}) \\ \qquad - 2\Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}), & i \neq j \\ \sum_{q=1}^{n_t} \Psi(\mathbf{x}_{ti}, \mathbf{x}_{tq}, \Omega_{ti})^2 \\ \qquad - 2\Psi(\mathbf{x}_{ti}, \mathbf{x}_{ti}, \Omega_{ti}) + 1, & i = j \end{cases} \tag{16}$$

Denote $\widehat{\mathbf{L}}^{I_2} \in \mathbb{R}^{n_t \times n_t}$ with

$$\left(\widehat{\mathbf{L}}^{I_2} \right)_{ij} = \begin{cases} -\Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}), & i \neq j \\ 1 - \Psi(\mathbf{x}_{ti}, \mathbf{x}_{tj}, \Omega_{ti}), & i = j \end{cases} \tag{17}$$

$\widehat{\mathbf{L}}^{I_2}$ is symmetric, and we have $\mathbf{L}^{I_2} = \widehat{\mathbf{L}}^{I_2} \widehat{\mathbf{L}}^{I_2 \top}$.

Denote $\mathbf{Z}_t = \mathbf{A}^\top \mathbf{X}_t$, and noticing the Eq.(11), we have

$$\begin{aligned} &\mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}_t \mathbf{L}^{I_2} \mathbf{X}_t{}^\top \mathbf{A} \right) \\ &= \mathrm{Tr}\left(\left(\mathbf{Z}_t \widehat{\mathbf{L}}^{I_2} \right) \left(\widehat{\mathbf{L}}^{I_2 \top} \mathbf{Z}_t^\top \right) \right) \\ &= \mathrm{Tr}\left(\mathbf{Z}^* \mathbf{Z}^{*\top} \right) \\ &= \sum_{i=1}^{n_t} \left\| \mathbf{A}^\top \mathbf{x}_{ti} - \mathbf{m}_t(\mathbf{x}_{ti}, \Omega_{ti}) \right\|^2 \end{aligned} \tag{18}$$

where $\mathbf{Z}^* = \left[\mathbf{z}_{t1} - \mathbf{m}_t(\mathbf{x}_{t1}, \Omega_{t1}) \,, \ldots, \, \mathbf{z}_{tn_t} - \mathbf{m}_t(\mathbf{x}_{tn_t}, \Omega_{tn_t}) \right]$.

According to the Eq.(15) and Eq.(18), the theorem is proved. ■

According to Proposition 1 and Theorem 1, the relationships between samples to samples in the process of class discriminative feature learning are further described by interpretable graph structures. The source and target class discriminative items Eq.(4) and Eq.(12) can be integrated to a unified form:

$$\min_{\mathbf{A}} \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X} \mathbf{L}^I \mathbf{X}^\top \mathbf{A} \right) \tag{19}$$

where $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t] \in \mathbb{R}^{d \times n}$, and $\mathbf{L}^I$ is the class discriminative Laplace matrix, $\mathbf{L}^I = \begin{bmatrix} \mathbf{L}^{I_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}^{I_2} \end{bmatrix}$.

C. Cross-Domain Alignment With Probability Information

Besides ensuring that the learned transfer features are sufficiently discriminative, aligning the geometric structure of two domains is another important consideration. Aiming to achieve the better cross-domain alignment, PGCD utilizes graph embedding structure to encourage the same-category samples in two domains to be close to each other in projection space. To this end, global and local cross-domain Laplacian matrices are proposed to increase the similarity of the same-category projected samples between two domains to realize the global and local structure alignment.

1) Global Cross-Domain Alignment: In order to align the global geometric structures, we propose a global cross-domain alignment item that considers the alignment between all samples of the same class in two domains. This item allows simultaneously bidirectional alignment of two domains.

To this end, we introduce the same-category target centroid of a source sample as follows:

$$\mathbf{m}_t(y_{si}) = \sum_{\mathbf{x}_{tj} \in \mathcal{D}_t^{y_{si}}} \mathrm{g}(\mathbf{x}_{tj}, y_{si}) \mathbf{A}^\top \mathbf{x}_{tj} \tag{20}$$

where $\mathrm{g}(\mathbf{x}_{tj}, y_{si})$ is the confidence weight defined in Eq.(3).

Firstly, to align the source samples with the target domain, we minimize the distance between each source sample and its same-category target centroid to make the source samples close to the same-category target samples.

$$\min_{\mathbf{A}} \sum_{i=1}^{n_s} \left\| \mathbf{A}^\top \mathbf{x}_{si} - \mathbf{m}_t(y_{si}) \right\|^2 \tag{21}$$

Secondly, in the process of aligning the target sample with the source domain, we minimize the distance between each target sample and its weighted source centroid $\mathbf{w}(\mathbf{x}_{tj}, \Omega_{tj})$:

$$\min_{\mathbf{A}} \sum_{j=1}^{n_t} \left\| \mathbf{A}^\top \mathbf{x}_{tj} - \mathbf{w}(\mathbf{x}_{tj}, \Omega_{tj}) \right\|^2 \tag{22}$$

where $\mathbf{w}(\mathbf{x}_{tj}, \Omega_{tj})$ is calculated by:

$$\mathbf{w}(\mathbf{x}_{tj}, \Omega_{tj}) = \sum_{c \in \Omega_{tj}} \frac{n_s^c \, \mathrm{g}(\mathbf{x}_{tj}, c)}{\sum_{c \in \Omega_{tj}} n_s^c \, \mathrm{g}(\mathbf{x}_{tj}, c)} \mathbf{m}_s(c) \tag{23}$$

with $\mathbf{m}_s(c) = \frac{1}{n_s^c} \sum_{\mathbf{x}_{si} \in \mathcal{D}_s^c} \mathbf{A}^\top \mathbf{x}_{si}$ being the source centroid of the c-th class, Ω_{tj} is the category aggregation of $\mathbf{x}_{tj}$, and $\frac{n_s^c \, \mathrm{g}(\mathbf{x}_{tj}, c)}{\sum_{c \in \Omega_{tj}} n_s^c \, \mathrm{g}(\mathbf{x}_{tj}, c)}$ is the weight of source centroid.

Theorem 2: The global cross-domain alignment item can be expressed as a graph embedding structure:

$$\begin{aligned}
&\sum_{i=1}^{n_s} \left\| \boldsymbol{A}^\top \boldsymbol{x}_{si} - \boldsymbol{m}_t(y_{si}) \right\|^2 \\
&\quad + \sum_{j=1}^{n_t} \left(\sum_{p=1}^{n_s} \mathrm{g}(\boldsymbol{x}_{tj}, y_{sp}) \right)^2 \left\| \boldsymbol{A}^\top \boldsymbol{x}_{tj} - \boldsymbol{w}(\boldsymbol{x}_{tj}, \Omega_{tj}) \right\|^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \boldsymbol{A}^\top \boldsymbol{x}_i - \boldsymbol{A}^\top \boldsymbol{x}_j \right\|^2 W_{ij}^G
\end{aligned} \tag{24}$$

where $\left(\sum_{p=1}^{n_s} \mathrm{g}(\boldsymbol{x}_{tj}, y_{sp}) \right)^2$ *is the weight for Eq.(22), and*

$$W_{ij}^G = \begin{cases}
-\sum_{p=1}^{n_t} \mathrm{g}(\boldsymbol{x}_{tp}, y_{si}) \mathrm{g}(\boldsymbol{x}_{tp}, y_{sj}), & i, j \le n_s, \; i \ne j \\
-\sum_{p=1}^{n_s} \mathrm{g}(\boldsymbol{x}_{t(i-n_s)}, y_{sp}) \mathrm{g}(\boldsymbol{x}_{t(j-n_s)}, y_{sp}), & i, j > n_s, \; i \ne j \\
\mathrm{g}(\boldsymbol{x}_{t(j-n_s)}, y_{si}) + \mathrm{g}(\boldsymbol{x}_{t(j-n_s)}, y_{si}) \sum_{p=1}^{n_s} \mathrm{g}(\boldsymbol{x}_{t(j-n_s)}, y_{sp}), & i \le n_s < j \\
\mathrm{g}(\boldsymbol{x}_{t(i-n_s)}, y_{sj}) + \mathrm{g}(\boldsymbol{x}_{t(i-n_s)}, y_{sj}) \sum_{p=1}^{n_s} \mathrm{g}(\boldsymbol{x}_{t(i-n_s)}, y_{sp}), & j \le n_s < i \\
0, & i = j
\end{cases} \tag{25}$$

Proof: Given W_{ij}^G, the adjacency matrix $\mathbf{W}^G$ is a symmetric matrix. By denoting $\mathbf{z}_i = \mathbf{A}^\top \mathbf{x}_i$, $\mathbf{z}_j = \mathbf{A}^\top \mathbf{x}_j$, we have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \mathbf{A}^\top \mathbf{x}_i - \mathbf{A}^\top \mathbf{x}_j \right\|^2 W_{ij}^G = 2 \, \mathrm{Tr} \left(\mathbf{A}^\top \mathbf{X} \mathbf{L}^G \mathbf{X}^\top \mathbf{A} \right) \tag{26}$$

where $\mathbf{L}^G \in \mathbb{R}^{n \times n}$ is a Laplacian matrix, $\mathbf{L}^G = \mathbf{D}^G - \mathbf{W}^G$, and $\mathbf{D}^G$ is a diagonal matrix with $D_{ii}^G = \sum_j W_{ij}^G$,

$$(\mathbf{L}^G)_{ij} = \begin{cases}
\sum_{p=1}^{n_t} \mathrm{g}(\mathbf{x}_{tp}, y_{si}) \mathrm{g}(\mathbf{x}_{tp}, y_{sj}), & i, j \le n_s, \; i \ne j \\
\sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t(i-n_s)}, y_{sp}) \mathrm{g}(\mathbf{x}_{t(j-n_s)}, y_{sp}), & i, j > n_s, \; i \ne j \\
-\mathrm{g}(\mathbf{x}_{t(j-n_s)}, y_{si}) - \mathrm{g}(\mathbf{x}_{t(j-n_s)}, y_{si}) \sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t(j-n_s)}, y_{sp}), & i \le n_s < j \\
-\mathrm{g}(\mathbf{x}_{t(i-n_s)}, y_{sj}) - \mathrm{g}(\mathbf{x}_{t(i-n_s)}, y_{sj}) \sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t(i-n_s)}, y_{sp}), & j \le n_s < i \\
1 + \sum_{p=1}^{n_t} \mathrm{g}(\mathbf{x}_{tp}, y_{si})^2, & i, j \le n_s, \; i = j \\
\left(\sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t(i-n_s)}, y_{sp}) \right)^2 + \sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t(i-n_s)}, y_{sp})^2, & i, j > n_s, \; i = j
\end{cases} \tag{27}$$

Denote

$$\widehat{\mathbf{L}}^G = \begin{bmatrix} \widehat{\mathbf{L}}^{d1} & \widehat{\mathbf{L}}^{g} \\ \widehat{\mathbf{L}}^{g\top} & \widehat{\mathbf{L}}^{d2} \end{bmatrix} \tag{28}$$

where $\widehat{\mathbf{L}}^{d1} = diag(1, 1, \ldots, 1)$, $\widehat{\mathbf{L}}^{d2} = diag\Big(\sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t1}, y_{sp}), \sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{t2}, y_{sp}), \ldots, \sum_{p=1}^{n_s} \mathrm{g}(\mathbf{x}_{tn_t}, y_{sp}) \Big)$, and

$$\widehat{\mathbf{L}}^{g} = \begin{bmatrix}
-\mathrm{g}(\mathbf{x}_{t1}, y_{s1}) & \cdots & -\mathrm{g}(\mathbf{x}_{tn_t}, y_{s1}) \\
-\mathrm{g}(\mathbf{x}_{t1}, y_{s2}) & \cdots & -\mathrm{g}(\mathbf{x}_{tn_t}, y_{s2}) \\
\vdots & \cdots & \vdots \\
-\mathrm{g}(\mathbf{x}_{t1}, y_{sn_s}) & \cdots & -\mathrm{g}(\mathbf{x}_{tn_t}, y_{sn_s})
\end{bmatrix}_{n_s \times n_t} \tag{29}$$

From Eq.(27) and Eq.(28), we can obtain $\mathbf{L}^G = \widehat{\mathbf{L}}^G \widehat{\mathbf{L}}^{G\top}$, and we can further get the following formula by combining

Eq.(20) and Eq.(23)

$$\begin{aligned}&\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{X}\mathbf{L}^G\mathbf{X}^\top\mathbf{A}\right)\\&=\mathrm{Tr}\left(\left(\mathbf{A}^\top\mathbf{X}\widehat{\mathbf{L}}^G\right)\left(\widehat{\mathbf{L}}^{G\top}\mathbf{X}^\top\mathbf{A}\right)\right)\\&=\sum_{i=1}^{n_s}\left\|\mathbf{A}^\top\mathbf{x}_{si}-\mathbf{m}_t(y_{si})\right\|^2\\&\quad+\sum_{j=1}^{n_t}\left\|\left(\sum_{p=1}^{n_s}\mathrm{g}(\mathbf{x}_{tj},y_{sp})\right)\mathbf{A}^\top\mathbf{x}_{tj}-\widehat{\mathbf{w}}(\mathbf{x}_{tj},\Omega_{tj})\right\|^2\end{aligned}\tag{30}$$

where $\widehat{\mathbf{w}}(\mathbf{x}_{tj},\Omega_{tj})=\sum_{p=1}^{n_s}\mathrm{g}(\mathbf{x}_{tj},y_{sp})\mathbf{A}^\top\mathbf{x}_{sp}$.

Since $\mathbf{w}(\mathbf{x}_{tj},\Omega_{tj})=\widehat{\mathbf{w}}(\mathbf{x}_{tj},\Omega_{tj})/\left(\sum_{p=1}^{n_s}\mathrm{g}(\mathbf{x}_{tj},y_{sp})\right)$, we have

$$\begin{aligned}&\sum_{j=1}^{n_t}\left\|\left(\sum_{p=1}^{n_s}\mathrm{g}(\mathbf{x}_{tj},y_{sp})\right)\mathbf{A}^\top\mathbf{x}_{tj}-\widehat{\mathbf{w}}(\mathbf{x}_{tj},\Omega_{tj})\right\|^2\\&=\sum_{j=1}^{n_t}\left(\sum_{p=1}^{n_s}\mathrm{g}(\mathbf{x}_{tj},y_{sp})\right)^2\left\|\mathbf{A}^\top\mathbf{x}_{tj}-\mathbf{w}(\mathbf{x}_{tj},\Omega_{tj})\right\|^2\end{aligned}\tag{31}$$

By the Eq.(26), Eq.(30) and Eq.(31), the theorem is proved. ∎

In Eq.(24), we set a weight $\left(\sum_{p=1}^{n_s}\mathrm{g}(\mathbf{x}_{tj},y_{sp})\right)^2$ for the distance between each target sample and its weighted source centroid. This weight reflects the influence of all source samples on the target sample $\mathbf{x}_{tj}$. The larger the weight, the greater the impact on the distance between $\mathbf{x}_{tj}$ and its weighted centroid.

Above all, the relationships between samples to samples in the process of domain align are further described by interpretable graph structures, the global cross-domain alignment item can be transformed as

$$\min_{\mathbf{A}}\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{X}\mathbf{L}^G\mathbf{X}^\top\mathbf{A}\right)\tag{32}$$

where the global cross-domain Laplacian matrix $\mathbf{L}^G$ is defined in Eq.(27).

2) Local Cross-Domain Alignment: PGCD minimizes the distance between each source sample and the same-category nearest neighbors in target domain to align the local structure in the projection space. The proposed local cross-domain alignment item is

$$\min_{\mathbf{A}}\sum_{\mathbf{x}_{si}\in\mathcal{D}_s}\sum_{\mathbf{x}_{tj}\in\mathcal{D}_t}\left\|\mathbf{A}^\top\mathbf{x}_{si}-\mathbf{A}^\top\mathbf{x}_{tj}\right\|^2W_{ij}^l\tag{33}$$

with

$$W_{ij}^l=\begin{cases}\exp\left(-\dfrac{\left\|\mathbf{x}_{si}-\mathbf{x}_{tj}\right\|^2}{2\eta}\right), & \mathbf{x}_{si}\in\mathcal{D}_s^c,\ \mathbf{x}_{tj}\in\mathcal{N}_t^c,\\ & c=1,2,\ldots C\\ 0, & \text{otherwise}\end{cases}\tag{34}$$

where $\mathcal{N}_t^c$ is a nearest neighbor set which contains h nearest target neighbors of $\mathbf{x}_{si}$ in the c-th class. The parameter η of heat kernel function and h are set as 1 and 5 in our experiments, respectively. Further, the local cross-domain alignment item Eq.(33) can be transformed as

$$\min_{\mathbf{A}}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|\mathbf{A}^\top x_i-\mathbf{A}^\top x_j\right\|^2\cdot W_{ij}^L=\min_{\mathbf{A}}2\,\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{X}\mathbf{L}^L\mathbf{X}^\top\mathbf{A}\right)\tag{35}$$

where $\mathbf{X}=[\mathbf{X}_s,\mathbf{X}_t]\in\mathbb{R}^{d\times n}$, $\mathbf{W}^L$ is adjacency matrix, and $\mathbf{W}^L=\begin{bmatrix}\mathbf{0} & \mathbf{W}^l\\ (\mathbf{W}^l)^T & \mathbf{0}\end{bmatrix}$ with the element W_{ij}^l defined by Eq.(34). $\mathbf{L}^L$ is the local cross-domain Laplacian matrix and $\mathbf{L}^L=\mathbf{D}^L-\mathbf{W}^L$, $\mathbf{D}^L$ is a diagonal matrix with $D_{ii}^L=\sum_j W_{ij}^L$.

In addition, we only align each source sample with its same-category target nearest neighbors for preserving the local spatial relationship without continuing to find source nearest neighbors for each target sample. This is because the category of the target sample is inaccurate, and finding the same-category source nearest neighbors for each target sample may cause the wrong alignment for the target sample and different-category source neighbors, which will hurt the local spatial relationship.

Above all, the cross-domain alignment item contains global cross-domain item Eq.(32) and local cross-domain item Eq.(35) can be represented as

$$\min_{\mathbf{A}}\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{X}(\mathbf{L}^G+\mathbf{L}^L)\mathbf{X}^\top\mathbf{A}\right)\tag{36}$$

D. Probability-Based Domain Distribution Alignment

In order to align the distributions of two domains, PGCD also embeds the probability-based domain distribution alignment item into the unified graph framework.

For aligning the marginal distribution across two domains, MMD-based method minimizes the distance between two domain centers as follows

$$\begin{aligned}&\min_{\mathbf{A}}\left\|\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{A}^\top\mathbf{x}_{si}-\frac{1}{n_t}\sum_{j=1}^{n_t}\mathbf{A}^\top\mathbf{x}_{tj}\right\|^2\\&\qquad=\min_{\mathbf{A}}\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{X}\mathbf{M}_0\mathbf{X}^\top\mathbf{A}\right)\end{aligned}\tag{37}$$

where the $\mathbf{M}_0\in\mathbb{R}^{n\times n}$ is a Laplacian matrix [9].

For aligning the conditional distribution across two domains, unlike the usual MMD-based method, probability information is used to construct the conditional distribution alignment item [13]:

$$\begin{aligned}&\min_{\mathbf{A}}\sum_{c=1}^{C}\left\|\frac{1}{n_s^c}\sum_{\mathbf{x}_{si}\in\mathcal{D}_s^c}\mathbf{A}^\top\mathbf{x}_{si}-\mathbf{m}_t(c)\right\|^2\\&\qquad=\min_{\mathbf{A}}\sum_{c=1}^{C}\mathrm{Tr}\left(\mathbf{A}^\top\mathbf{X}\mathbf{M}_c\mathbf{X}^\top\mathbf{A}\right)\end{aligned}\tag{38}$$

where $\mathbf{m}_t(c)$ the target centroid of the c-th class is defined in Eq.(8).

We find the matrix $\mathbf{M}_c\in\mathbb{R}^{n\times n}$ is a Laplacian matrix, and $\mathbf{M}_c=\mathbf{D}_c-\mathbf{W}_c$, $\mathbf{D}_c$ is a diagonal matrix with $D_{cii}=\sum_j W_{cij}$.

The adjacency matrix $\mathbf{W}_c$ is

$$(\mathbf{W}_c)_{ij} = \begin{cases} -1/n_s^c n_s^c, & \mathbf{x}_{si}, \mathbf{x}_{sj} \in \mathcal{D}_s^c,\ i \neq j \\ -\mathrm{g}(\mathbf{x}_{ti}, c)\,\mathrm{g}(\mathbf{x}_{tj}, c), & \mathbf{x}_{ti}, \mathbf{x}_{tj} \in \mathcal{D}_t^c,\ i \neq j \\ \mathrm{g}(\mathbf{x}_{tj}, c)/n_s^c, & \mathbf{x}_{si} \in \mathcal{D}_s^c,\ \mathbf{x}_{tj} \in \mathcal{D}_t^c \\ \mathrm{g}(\mathbf{x}_{ti}, c)/n_s^c, & \mathbf{x}_{ti} \in \mathcal{D}_t^c,\ \mathbf{x}_{sj} \in \mathcal{D}_s^c \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

The probability-based joint distribution alignment item that contains marginal and conditional distribution alignment items can be expressed as follows:

$$\min_{\mathbf{A}} \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}\mathbf{M}\mathbf{X}^\top \mathbf{A}\right) \tag{40}$$

where $\mathbf{M} = \left(\mathbf{M}_0 + \sum_{c=1}^{C} \mathbf{M}_c\right)$, and $\mathbf{M}$ is a Laplacian matrix.

E. Overall Formulation and Optimization

By incorporating Eq.(19), Eq.(36), and Eq.(40), PGCD can be formulated as

$$\begin{aligned} &\min_{\mathbf{A}}\ \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{X}\Big(\gamma\mathbf{M} + \beta\mathbf{L}^I + \alpha(\mathbf{L}^G + \mathbf{L}^L)\Big)\mathbf{X}^\top \mathbf{A}\right) + \lambda \left\|\mathbf{A}\right\|_F^2 \\ &s.t.\ \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}_z \end{aligned} \tag{41}$$

where $\mathbf{I}_z$ is an identity matrix of dimension z and the data centering matrix $\mathbf{H} = \mathbf{I}_n - (1/n)\mathbf{1}_n$. Parameters α, β and γ are trade-off parameters with the positive values. The regularization term $\lambda \left\|\mathbf{A}\right\|_F^2$ tries to avoid numerical instability issue, with respect to λ, small values could cause trivial solutions, large values could have a greater impact on the loss function. Furthermore, $\mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}_z$ is proposed to avoid trivial solutions and maximize the variances for embedded source and target data.

Clearly, the constrained optimization problem Eq.(41) can be derived by the following Lagrange function:

$$\begin{aligned} \mathcal{L}(\mathbf{A}, \Phi) = &\mathrm{Tr}\left(\mathbf{A}^\top \Big(\mathbf{X}\big(\gamma\mathbf{M} + \beta\mathbf{L}^I + \alpha(\mathbf{L}^G + \mathbf{L}^L)\big)\mathbf{X}^\top + \lambda\mathbf{I}_d\Big)\mathbf{A}\right) \\ &+ \mathrm{Tr}\left(\Phi(\mathbf{I}_z - \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A})\right) \end{aligned} \tag{42}$$

where $\Phi = \mathrm{diag}\,(\phi_1, \phi_2, \ldots, \phi_z) \in \mathbb{R}^{z\times z}$ is diagonal matrix with Lagrange multipliers. In Eq.(42), by setting the derivative of projection matrix $\mathbf{A}$ to zero, the generalized eigenvalue problem will be obtained as

$$\Big(\mathbf{X}\Big(\gamma\mathbf{M} + \beta\mathbf{L}^I + \alpha(\mathbf{L}^G + \mathbf{L}^L)\Big)\mathbf{X}^\top + \lambda\mathbf{I}_d\Big)\mathbf{A} = \Phi\mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A} \tag{43}$$

The optimal solution is the generalized eigenvector matrix corresponding to the first z smallest eigenvectors. With the learned projection matrix $\mathbf{A}$, the new projection feature matrix $\mathbf{Z} = \mathbf{A}^\top\mathbf{X} \in \mathbb{R}^{z\times n}$ can be obtained for the classification task.

Base on the above discussion, the proposed algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed Algorithm PGCD

Input: Source data and labels$\{\mathbf{X}_s, \mathbf{y}_s\}$; target data$\mathbf{X}_t$; projection space dimension z; iterations T; cross-domain nearest neighbors h; the soft label nearest neighbors p; trade-off parameters α, β, λ, γ.

1. Initialize $\sum_{c=1}^{C}\mathbf{M}_c$, $\mathbf{L}^G$, $\mathbf{L}^L$, $\mathbf{L}^{I_2}$ as 0.
2. Construct $\mathbf{M}_0$ by Eq.(37); construct $\mathbf{L}^{I_1}$ by Eq.(6).
3. While not converged and iteration $\leq T$ **do**
4. Take the z smallest eigenvectors to solve the generalized eigen-decomposition problem by Eq.(43) for new projection feature matrix $\mathbf{A}$.
5. Obtain new projection feature matrix $\mathbf{Z}$.
6. Train a 1-NN classifier $\mathcal{F}$ for classification task of target domain.
7. Train a soft label K-NN classifier $\mathcal{F}_{soft}$.
8. Generate soft labels and confidence weights by Eq.(1) and Eq.(3).
9. Update $\sum_{c=1}^{C}\mathbf{M}_c$ by Eq.(39).
10. Update class discriminative Laplacian matrix $\mathbf{L}^I$ by Eq.(19).
11. Update global cross-domain Laplacian matrix $\mathbf{L}^G$ by Eq.(27).
12. Update local cross-domain Laplacian matrix $\mathbf{L}^L$ by Eq.(34) and Eq.(35).
13. End while
Output: Projection matrix $\mathbf{A}$; classifier $\mathcal{F}$.

TABLE I
INFORMATION OF USED CROSS-DOMAIN DATASETS IN EXPERIMENTS

Dataset	Type	Feature	Class	Sample	Domain
Office-10 (SURF / DeCAF$_6$)	Object	800/4096	10	1410	A,W,D
Caltech-10 (SURF / DeCAF$_6$)	Object	800/4096	10	1123	C
Office-31 (DeCAF$_7$ / Resnet50)	Object	4096/2048	31	4110	A,W,D
Office-Home (Resnet50)	Object	2048	65	15585	Ar,Cl,Pr,Re
CMU-PIE	Face	1024	68	11554	C05,C07 C09,C27,C29

F. Algorithm Complexity Analysis

In this part, we analyze the complexity of Algorithm 1. In step 2 and step 9, the cost of constructing MMD matrix $\mathbf{M}$ is $\mathcal{O}\left(C\,(n_s + n_t)^2 + (n_s + n_t)^2\right)$. In step 10, the cost of calculating the class discriminative Laplacian matrix $\mathbf{L}^I$ is $\mathcal{O}(C(n_s^2 + n_t^2))$. In step 11 and step 12, the cost of calculating cross-domain Laplacian matrix $\mathbf{L}^G$ and $\mathbf{L}^L$ is $\mathcal{O}(C(n_s n_t))$. Solving the generalized eigen-decomposition problem costs $\mathcal{O}(z\,d^2)$. The remaining steps will cost around $\mathcal{O}(d(n_s + n_t))$. The number of iterations is T and $n_s + n_t = n$. Therefore, the computational complexity of PGCD is $\mathcal{O}\left(TC(n^2 + n_s^2 + n_t^2 + n_s n_t) + Tdn + Tzd^2\right)$.

IV. EXPERIMENTS

A. Experimental Datasets

As shown in TableI, multiple benchmark datasets are used in experiments, including Office-Caltech-10 (SURF / DeCAF$_6$),

Office-31 (DeCAF_7 / ResNet50), Office-Home (ResNet50), and CMU-PIE.[1]

Office-Caltech-10: It contains 1123 images from dataset Caltech-256 (C) and 1410 images from Office-31 with 10 common categories. The 800-dim SURF features and the 4096-dim DeCAF_6 features are used for experiments. DeCAF_6 features are collected from the first fully connected layer of the AlexNet.

Office-31: It contains 4110 images of 31 object categories and 3 domains: Amazon (A), DSLR (D), and Webcam (W). Specifically, the Amazon dataset contains images from the online web, the DSLR dataset contains high-resolution images shot by digital SLR cameras, and the Webcam dataset contains low-resolution images shot by web cameras. We use the 4096-dim DeCAF_7 feature collected by the AlexNet, and 2048-dim feature collected by the ResNet50.

Office-Home: It contains 15585 images of 65 object categories and 4 domains: Artistic images (Ar), Clip images (Cl), Product images (Pr), and Real-World images (Rw). The 2048-dim ResNet50 features are adopted for experiments.

CMU-PIE: It contains 11554 facial images of 68 people and 5 domains: C05 (left), C07 (upward), C09 (downward), C27 (front) and C29 (right). These domains represent 5 different face angles, and the image size in each domain is 32×32.

B. Experimental Setup

In order to evaluate the effectiveness of the proposed algorithm, we compare PGCD with standard learning methods, shallow UDA methods, and deep UDA methods.

Standard learning methods include: 1NN and SVM.

Shallow UDA methods include: GFK [48], TCA [7], JDA [8], DTSL [22], RTML [28], JGSA [24], DICD [11], LDADA [27], DICE [3], PACET [13], MLOT [49], GEF-LDA [9], HCA [25], ACE [17], MEDA [26], and RTFL [29].

Deep UDA methods include: AlexNet [50], ResNet [51], DDC [30], DAN [31], DANN [34], RTN [52], DCORAL [53], DUCDA [54], JAN [32], CDAN [35], TAT [36], GAACN [37], GADSN [38], AEO [39], DCAN [45], GVB-GD [40], RSDA [41], GSDA [33], SRDC [42] and FixBi [43].

In all experiments, we fix the cross-domain nearest neighbors $h = 5$, the soft label nearest neighbors $p = 5$, iterations $T = 20$, set $\lambda = 0.1$ for Office-Caltech-10 and Office31, set $\lambda = 0.01$ for Office-Home and CMU-PIE, set $\gamma = 1$ for all datasets except Office31 that we set $\gamma = 5$. Besides, for α, we set $\alpha = 1$ for Office-Caltech-10 (SURF) and Office-Home, set $\alpha = 0.1$ for Office-Caltech-10 (DeCAF_6) and Office31, set $\alpha = 5$ for CMU-PIE. For β, we set $\beta = 1$ for Office-Caltech-10, Office-Home and Office31 (Resnet50), set $\beta = 5$ for Office31 (DeCAF_7), set $\beta = 0.01$ for CMU-PIE.

For a fair comparison, we choose 1NN as the classifier to predict the classification accuracy like other methods. Moreover, experimental results of comparison methods are used from their original articles if the experimental protocol is identical with ours. The classification accuracy of the target data is proposed to be the evaluation metric, which is defined as

$$\text{Accuracy} = \frac{|\mathbf{x} : \mathbf{x} \in D_t \wedge \widehat{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} : \mathbf{x} \in D_t|} \tag{44}$$

where D_t is the target domain, $y(\mathbf{x})$ is the truth label of x, and $\widehat{y}(\mathbf{x})$ is the predicted label. The code is available at https://github.com/csrwang/PGCD.

C. Experimental Results and Discussion

To examine the effectiveness of the proposed PGCD, we compare it with extensive relevant UDA methods on the popular benchmark datasets.

1) Results on the Office-Caltech-10 Dataset: In the Office-Caltech-10 dataset, low-dimensional features SURF and high-dimensional deep features DeCAF_6 are used in the experiments. As shown in Table II, PGCD achieves the best average result of classification accuracy and also has the best results on 4 out of 12 cross-domain tasks with SURF features. Methods ACE, HCA, PACET, MEDA and DICE have achieved average accuracy results of more than 51%, but as a contrast, the average result of PGCD still has a clear lead.

As shown in Table III, PGCD achieves the third place average accuracy 91.2% and MEDA has the best average accuracy with DeCAF_6 features. Through carefully comparing the results on SURF and DeCAF_6 features, we can find that the average accuracy of PGCD is 2.2% higher than that of MEDA on SURF; the average accuracy of MEDA is 1.6% higher than that of PGCD on DeCAF_6. Nevertheless, PGCD can achieve good performance on two features, which illustrates that PGCD has a good generalization capacity.

2) Results on the Office-31 Dataset: Table IV and Table V show the experimental results of the proposed method PGCD compared with shallow and deep UDA methods in Office-31 (DeCAF_7) respectively. Compared with shallow UDA methods, PGCD achieves the best average accuracy and has the best results on 4 out of 6 tasks. Compared with deep UDA methods, PGCD has the best average result and has great performance on 3 tasks D→A, W→A, and D→W. In cross-domain task D→A and W→A, PGCD gains great performance improvements compared with the shallow and deep UDA methods. It is noteworthy that PGCD has the best result on the task which transfers from a small sample size source domain to a large sample size target domain, such as D→A and W→A. It indicates that PGCD can make full use of cross-domain information and fully align the cross-domain geometry when the source domain provides limited knowledge. To sum up, PGCD can achieve great performance compared with shallow and deep UDA methods on Office-31 (DeCAF_7).

We used the Resnet50 as a feature extraction network to extract deep features of Office-31, and compared PGCD with the SOTA deep UDA methods. The experimental results are shown in Table VI. Compared with deep UDA methods, the average accuracy of PGCD is leading the average accuracies of GVB-GD, GSDA, and DCAN by 0.9%, 0.5%, and 0.7%, respectively. Moreover, PGCD achieves the same average accuracy as RSDA-DANN. To sum up, PGCD as a lightweight shallow UDA method can achieve better or comparable performance than most SOTA deep UDA methods on Office-31 (Resnet50).

[1]The datasets come from https://github.com/jindongwang/transferlearning/tree/master/data

TABLE II
ACCURACIES (%) ON THE OFFICE-CALTECH-10 DATASET WITH SURF FEATURES

Methods	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
1NN	23.7	25.8	25.5	26.0	29.8	25.5	19.9	23.0	59.2	26.3	28.5	63.4	31.4
SVM	50.1	43.1	47.8	42.8	37.0	37.2	29.5	34.2	80.6	30.1	32.1	72.2	44.7
GFK	41.0	40.7	38.9	40.3	38.9	36.3	30.7	29.8	80.9	30.3	32.1	75.6	43.0
TCA	38.2	38.6	41.4	37.8	37.6	33.1	29.3	30.1	87.3	31.7	32.2	86.1	43.6
JDA	44.8	41.7	45.2	39.4	38.0	39.5	31.2	32.8	89.2	31.5	33.1	89.5	46.3
DTSL	51.3	38.7	47.1	43.4	36.6	38.8	29.8	34.1	82.8	30.1	32.1	72.2	45.7
RTML	49.3	44.7	47.6	43.7	44.3	43.9	34.8	35.3	91.0	34.6	33.3	89.0	49.3
JGSA	51.5	45.4	45.9	41.5	45.8	47.1	33.2	39.9	90.5	29.9	38.0	**91.9**	50.1
DICD	47.3	46.4	49.7	42.4	45.1	38.9	33.6	34.1	89.8	34.6	33.3	89.0	49.3
LDADA	54.8	**60.2**	41.5	38.4	49.3	39.1	31.7	35.1	74.6	29.9	40.6	74.7	47.5
DICE	50.2	48.1	51.0	42.7	52.2	49.7	37.8	37.5	87.3	33.7	41.1	84.1	51.3
MEDA	**56.5**	53.9	50.3	43.9	53.2	45.9	34.0	42.7	88.5	34.9	41.2	87.5	52.7
PACET	52.2	51.5	52.2	44.3	**53.2**	50.3	**39.0**	40.8	**92.4**	34.5	40.8	91.5	53.6
MLOT	51.5	45.9	52.2	42.3	41.3	40.8	33.2	38.0	90.8	34.4	37.8	87.8	49.7
GEF-LDA	48.2	47.8	50.3	42.7	46.4	36.9	33.6	34.0	**92.4**	35.4	34.8	90.5	49.4
HCA	48.4	57.7	56.1	43.8	56.9	51.8	37.1	**46.7**	82.1	**37.1**	**46.5**	84.2	54.0
ACE	53.7	42.7	54.1	45.0	46.1	42.0	34.7	37.8	91.1	34.7	37.8	**91.9**	51.0
PGCD(ours)	54.9	56.6	**59.2**	**45.2**	50.5	**52.2**	36.9	41.1	**92.4**	36.1	44.1	89.2	**54.9**

TABLE III
ACCURACIES (%) ON THE OFFICE-CALTECH-10 DATASET WITH DeCAF$_6$ FEATURES

Methods	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
1NN	87.3	72.5	79.62	71.7	68.1	73.9	55.3	62.5	98.0	42.0	49.9	91.5	71.1
SVM	91.6	80.7	86.0	82.2	71.9	80.9	67.9	73.4	**100.0**	72.8	78.7	98.3	82.0
GFK	87.3	75.9	83.4	80.3	77.0	80.9	67.8	74.3	**100.0**	69.1	75.8	98.6	80.9
TCA	89.8	78.3	85.4	82.6	74.2	81.5	80.4	84.1	**100.0**	82.3	89.1	99.7	85.6
JDA	90.2	85.4	86.0	81.9	80.7	81.5	81.2	90.7	**100.0**	80.3	92.0	99.3	87.4
DTSL	91.5	76.6	87.9	85.8	73.6	82.2	72.8	75.5	**100.0**	75.2	85.0	99.3	83.8
RTML	90.6	85.4	89.3	86.4	80.3	84.4	83.3	91.4	**100.0**	85.7	91.9	99.0	89.0
JGSA	91.4	86.8	**93.6**	84.9	81.0	88.5	84.9	90.7	**100.0**	86.2	92.0	99.7	90.0
DICD	90.9	92.2	**93.6**	86.0	81.4	83.4	84.0	89.7	**100.0**	86.1	92.1	99.0	89.9
LDADA	92.5	86.4	88.0	**88.6**	**90.5**	85.0	87.0	92.0	96.8	86.6	90.9	95.0	89.9
DICE	92.3	93.6	**93.6**	85.9	86.4	89.8	85.3	90.7	**100.0**	87.4	92.5	99.0	91.4
MEDA	**93.4**	**95.6**	91.1	87.4	88.1	88.1	**93.2**	**99.4**	99.4	87.5	**93.2**	97.6	**92.8**
MLOT	91.3	81.0	78.0	84.7	81.4	78.2	83.3	91.9	92.2	84.4	90.1	95.1	86.0
GEF-LDA	91.2	89.2	88.5	83.6	76.3	82.2	84.0	89.0	**100.0**	86.3	92.2	99.0	88.5
ACE	92.1	89.5	88.5	84.5	87.8	86.0	84.5	92.3	**100.0**	86.3	91.9	**100.0**	90.3
PGCD(ours)	92.5	91.2	92.4	86.5	84.1	**90.4**	85.3	91.6	**100.0**	**87.6**	92.5	**100.0**	91.2

TABLE IV
ACCURACIES (%) ON THE OFFICE31 DATASET WITH DeCAF$_7$ FEATURES COMPARED WITH SHALLOW METHODS

Methods	A→D	A→W	D→A	D→W	W→A	W→D	Average
1NN	59.6	54.0	42.4	90.9	40.8	97.8	64.3
SVM	55.7	50.6	46.5	93.1	43.0	97.4	64.4
GFK	58.6	58.4	52.4	93.6	46.1	91.0	66.7
TCA	57.8	59.0	51.6	90.2	47.3	88.2	65.7
JDA	56.6	56.9	47.0	94.1	45.0	98.0	66.3
DTSL	60.0	54.5	46.6	94.3	45.6	94.0	65.8
JGSA	63.5	56.0	58.5	97.2	55.4	98.2	71.5
DICD	62.7	56.7	54.2	96.5	51.2	98.6	70.0
LDADA	68.0	63.0	56.2	83.7	55.0	93.6	69.9
DICE	66.7	71.4	56.5	96.9	58.6	99.8	75.0
PACET	69.1	**71.7**	62.3	97.4	59.2	**100.0**	76.6
MLOT	59.6	61.4	52.8	92.0	50.8	87.0	67.3
PGCD(ours)	**71.7**	68.8	**63.9**	**97.5**	**61.2**	99.0	**77.0**

TABLE V
ACCURACIES (%) ON THE OFFICE31 DATASET WITH DeCAF$_7$ FEATURES COMPARED WITH DEEP METHODS

Methods	A→D	A→W	D→A	D→W	W→A	W→D	Average
AlexNet	63.8	61.6	51.1	95.4	49.8	99.0	70.1
DDC	64.4	61.8	52.1	95.0	52.2	98.5	70.7
DAN	67.0	68.5	54.0	96.0	53.1	99.0	72.9
DANN	**72.3**	73.0	53.4	96.4	51.2	99.2	74.3
RTN	71.0	**73.3**	50.5	96.8	51.0	99.6	73.7
DCORAL	66.4	66.8	52.8	95.7	51.5	99.2	72.1
DUCDA	68.3	68.3	53.6	96.2	51.6	**99.7**	73.0
PGCD(ours)	71.7	68.8	**63.9**	**97.5**	**61.2**	99.0	**77.0**

3) Results on the Office-Home Datasets: As shown in Table VII, the proposed method PGCD is significantly better than other methods compared with shallow methods in Office-Home. Specifically, PGCD has the best results on 11 out of 12 cross-domain tasks and achieves the best average result 68.8% which is 2.6% higher than that of the second-best method DICD.

Moreover, Office-Home is a challenging benchmark dataset and widely used in experiments of deep methods. As shown in Table VIII, compared with the state-of-the-art deep UDA methods, our method can achieve the best average accuracy that leads the second-best method AEO by 1.0% and it has the best results on 7 out of 12 tasks compared with all the deep methods obviously. Although Office-Home has 65 categories and they are difficult to be distinguished, CDFP is still able to achieve better accuracy, which benefits from its ability of learning the discriminative features of classes while maintaining the similarity and consistency between domains.

TABLE VI
ACCURACIES (%) ON THE OFFICE31 DATASET WITH RESNET50 FEATURES COMPARED WITH DEEP METHODS

Methods	A→D	A→W	D→A	D→W	W→A	W→D	Average
GVB-GD	95.0	94.8	73.4	98.7	73.7	**100.0**	89.3
RSDA-DANN	95.2	95.3	75.5	**99.3**	76.0	**100.0**	90.2
GSDA	94.8	95.7	73.5	99.1	74.9	**100.0**	89.7
SRDC	**95.8**	95.7	76.7	99.2	77.1	**100.0**	90.8
FixBi	95.0	**96.1**	**78.7**	**99.3**	**79.4**	**100.0**	**91.4**
DCAN	92.6	95.0	77.2	97.5	74.9	**100.0**	89.5
PGCD(ours)	95.2	94.0	76.4	99.0	76.5	**100.0**	90.2

4) Results on the CMU-PIE Dataset: The experimental results carried out on the facial images dataset CMU-PIE can be observed in TableIX. Compared with the state-of-the-art approaches, the accuracy of PGCD has a significant improvement. PGCD has the best results on 16 out of 20 cross-domain tasks and achieves the best average accuracy 85.4% that is 4.4% higher than the second-best method PACET and leads other methods a lot.

Note that there are more than 60 categories in CMU-PIE and Office-Home. It is well known that the increase of categories will increase the difficulty of transfer feature learning as the feature distortion and mix in the projection space will be more serious. PGCD not only improves the average accuracy significantly but also performs well in most cross-domain tasks. This is because that it can effectively learn the discriminative transfer features, meanwhile, align the local and global cross-domain geometric structures to improve the classification performance.

D. Empirical Analysis

1) Ablation Study: In order to find out the contributions of different items in PGCD, the ablation study is carried out. As shown in TableX, 1) GCD represents PGCD without probability information; 2) PGD represents PGCD without cross-domain alignment item Eq.(36); 3) PGC represents PGCD without class discriminative item Eq.(19); 4) PD represents the class discriminative item Eq.(19); 5) PD_S represents the source class discriminative item Eq.(7); 6) PD_T represents the target class discriminative item Eq.(12); 7) PC_{L+G} represents the cross-domain alignment item Eq.(36); 8) PC_L represents the local cross-domain alignment item Eq.(35); 9) PC_G represents the global cross-domain alignment item Eq.(32); 10) PGCD_S represents PGCD without the target class discriminative item Eq.(12) (PD_T); 11) PGCD_T represents PGCD without the source class discriminative item Eq.(7) (PD_S); 12) PGC_LD represents PGCD without the global cross-domain alignment item Eq.(32) (PC_G); 13) PGC_GD represents PGCD without the local cross-domain alignment item Eq.(35) (PC_L); 14) $\text{PC}_{G(S\to T)}$ represents the source global cross-domain alignment item Eq.(21); 15) $\text{PC}_{G(T\to S)}$ represents the target global cross-domain alignment item Eq.(22); 16) PACET_{CDL} represents the cross-domain alignment item in [13] (Eq.(3) in [13]); 17) $\text{PC}_{L(T\to S)}$ represents the local cross-domain alignment item which finds the source nearest neighbors for the target domain (note that PGCD does not contain $\text{PC}_{L(T\to S)}$). In $\text{PC}_{L(T\to S)}$, the local cross-domain alignment item has the same form as Eq.(33) but with a different adjacency matrix, as follows:

$$W_{ij}^{l}=\begin{cases}\exp\left(-\dfrac{\left\|\mathbf{x}_{ti}-\mathbf{x}_{sj}\right\|^2}{2\eta}\right), & \mathbf{x}_{ti}\in\mathcal{D}_t^c,\ \mathbf{x}_{sj}\in\mathcal{N}_s^c,\\ & c=1,2,\ldots C\\ 0, & \text{otherwise}\end{cases}$$

We choose one cross-domain task in each dataset for the ablation study. 1) Comparing GCD, PGD, PGC, and PGCD, the average accuracy of PGC is the lowest (64.1%), PGD is the second-lowest (64.4%), and GCD is the relatively highest (65.8%). Whether the result of each task or the result of average accuracy, PGCD gets the highest results. Based on the above observations, each proposed item in PGCD and the probability information in each item have the contribution to the classification accuracy, and the class discriminative item Eq.(19) that contains the probability information contributes the most. In addition, the results of PGC and PGD in each task are closely related to the trade-off parameters α and β. For instance, in most tasks, the results of PGD are higher than those of PGC except task C05→C09. In task C05→C09, the accuracy of PGC leads PGD by 13%, because in this task the optimal trade-off parameter α is much bigger than β. To sum up, all the proposed items are indispensable and complement each other.

2) To verify the performance degrade when removing one of the two terms (PD_S, PD_T) in the class discriminative item PD (Eq.(19)) of PGCD, we experiment with PGCD_S and PGCD_T. Compared with PGCD, the average accuracy of PGCD_S drops by 1.8%, and the average accuracy of PGCD_T drops by 4.1%. Moreover, using only PD_S, the average accuracy is 52.4%; using only PD_T, the average accuracy is 51.4%; PD consisting of the above two terms can improve the average accuracy to 56.3%. So, the PD_S can improve the performance of the model by using the labeled source domain to learn the accurate discriminative features. The PD_T can reduce the uncertainty of pseudo labels and learn discriminative features on unlabeled target domains to improve model accuracy.

3) To verify the performance degrade when removing one of the two terms (PC_L, PC_G) in the cross-domain alignment item PC_{L+G} (Eq.(36)) of PGCD, we experiment with PGC_LD and PGC_GD. Compared with PGCD, the average accuracy of PGC_LD drops by 1.0%, and the average accuracy of PGC_GD drops by 1.8%. Moreover, using only PC_L, the average accuracy is 60.1%; using only PC_G, the average accuracy is 61.8%; PC_{L+G} consisting of the above two terms can improve the average accuracy to 62.5%. Therefore, using either PC_L or PC_G has a good performance improvement effect on PGCD. Moreover, they are beneficial to each other, and the combination can further improve the performance of the proposed model.

4) The target global cross-domain alignment item $\text{PC}_{G(T\to S)}$ makes each target sample close to its weighted source centroid. PACET_{CDL} in [13] makes each target sample close to the source center of the pseudo label class and far from the source center of the nearest negative class. Comparing the performance of the two strategies, we can see that $\text{PC}_{G(T\to S)}$ achieves significantly higher accuracy on each task, and the average accuracy of $\text{PC}_{G(T\to S)}$ is 10.0% higher than that of PACET_{CDL}. The reason for this result can be considered

TABLE VII
ACCURACIES (%) ON OFFICE-HOME DATASET WITH RESNET50 FEATURES COMPARED WITH SHALLOW METHODS

Methods	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
1NN	35.9	54.4	64.9	39.5	48.4	51.4	41.8	32.4	64.1	58.1	39.5	69.6	50.0
SVM	36.5	54.7	64.9	40.6	48.1	52.5	42.0	32.6	64.4	57.7	40.1	69.8	50.3
GFK	38.7	57.7	63.0	43.3	54.6	54.2	48.0	41.6	66.8	58.1	45.0	72.8	53.7
JDA	45.8	63.6	67.5	53.3	62.2	62.9	56.0	47.1	72.9	61.8	50.5	75.2	59.9
JGSA	50.3	70.0	73.8	52.7	68.9	68.2	55.6	47.9	75.1	64.0	52.0	78.7	63.1
DICD	53.0	73.6	75.7	**59.7**	70.3	70.6	60.9	49.4	77.7	67.9	56.2	79.7	66.2
DICE	53.2	72.4	74.5	56.5	70.1	69.1	58.9	51.5	77.0	66.5	54.8	79.0	65.3
HCA	51.2	69.8	76.2	53.7	67.1	66.5	55.9	49.0	76.6	67.6	55.7	81.1	64.2
ACE	32.6	49.2	54.9	36.8	51.1	51.2	35.7	32.7	58.3	47.4	39.5	65.9	46.3
PGCD(ours)	**57.7**	**77.2**	**79.1**	59.1	**74.3**	**72.7**	**61.2**	**54.2**	**79.3**	**70.0**	**58.4**	**82.7**	**68.8**

TABLE VIII
ACCURACIES (%) ON OFFICE-HOME DATASET WITH RESNET50 FEATURES COMPARED WITH DEEP METHODS

Methods	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
ResNet	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
TAT	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
GAACN	53.1	71.5	74.6	**59.9**	64.6	67.0	59.2	53.8	75.1	70.1	**59.3**	80.9	65.8
GADSN	51.7	71.5	75.7	51.9	66.1	68.6	53.9	49.0	74.0	64.0	53.2	78.6	63.2
AEO	52.2	73.6	76.9	59.7	72.1	**73.2**	**61.3**	52.1	78.9	**72.4**	58.1	82.6	67.8
PGCD(ours)	**57.7**	**77.2**	**79.1**	59.1	**74.3**	72.7	61.2	**54.2**	**79.3**	70.0	58.4	**82.7**	**68.8**

TABLE IX
ACCURACIES (%) ON CMU-PIE DATASETS

Methods	1NN	SVM	GFK	TCA	JDA	DTSL	RTML	JGSA	DICD	LDADA	PACET	DICE	GEF-LDA	ACE	RTFL	PGCD
C05→C07	26.1	30.9	26.2	40.8	58.6	65.9	60.1	52.9	73.0	34.5	82.2	84.1	81.3	68.8	61.3	**89.3**
C05→C09	26.6	33.9	27.3	41.8	52.0	64.1	55.2	53.1	72.0	44.9	80.8	77.9	77.3	62.4	61.6	**82.8**
C05→C27	30.7	41.4	31.2	59.6	83.7	82.0	85.2	66.0	92.2	61.5	94.9	**95.9**	94.0	84.9	83.8	95.0
C05→C29	16.7	23.8	17.6	29.4	47.7	54.9	53.0	46.1	**66.9**	35.4	64.5	66.5	55.4	54.4	42.9	66.0
C07→C05	24.5	31.8	25.2	41.8	60.6	45.0	58.1	57.5	69.9	31.4	82.9	81.3	73.7	66.8	59.8	**84.4**
C07→C09	46.6	41.0	47.4	51.5	60.2	53.5	63.9	57.2	65.9	34.9	73.5	74.0	76.5	72.7	71.2	**86.1**
C07→C27	54.1	62.2	54.3	64.7	75.4	71.4	76.2	69.2	85.3	53.5	90.2	88.6	89.2	85.5	82.9	**90.8**
C07→C29	26.5	28.8	27.1	33.7	40.9	48.0	40.4	49.8	48.7	26.4	**72.4**	68.8	64.2	52.0	56.7	71.1
C09→C05	21.4	32.3	21.8	34.7	50.9	52.5	53.1	56.1	69.4	38.2	79.7	78.8	69.3	64.5	55.1	**83.8**
C09→C07	41.0	39.7	43.2	47.7	56.1	55.6	58.7	58.7	65.4	30.5	79.3	76.7	70.7	69.4	68.3	**83.1**
C09→C27	46.5	61.9	46.4	56.2	68.0	77.5	69.8	69.5	83.4	60.6	84.6	85.2	86.1	87.9	82.7	**95.5**
C09→C29	26.2	37.7	26.7	33.2	40.3	54.1	42.1	52.2	61.4	40.7	70.2	70.8	69.3	62.3	58.3	**78.2**
C27→C05	33.0	57.7	34.2	55.6	81.0	81.5	81.1	63.2	93.1	61.3	94.0	93.3	93.9	88.2	82.0	**98.1**
C27→C07	52.7	69.2	62.9	67.8	82.8	85.4	83.9	65.7	90.1	56.7	93.5	95.0	94.5	91.7	88.6	**95.5**
C27→C09	73.2	69.7	73.4	75.9	87.2	82.2	89.5	62.6	89.0	67.8	91.3	92.3	92.0	92.6	89.4	**93.0**
C27→C29	37.2	48.7	37.4	40.3	49.9	72.6	56.3	57.0	75.6	50.4	76.9	81.1	81.9	77.3	74.8	**84.5**
C29→C05	18.5	29.4	20.4	27.0	47.	52.2	29.1	56.3	62.9	31.3	76.2	73.8	55.7	55.3	46.0	**82.1**
C29→C07	24.2	33.1	24.6	30.0	44.8	49.4	33.3	54.7	57.0	24.1	69.2	71.2	64.5	54.5	57.6	**80.7**
C29→C09	28.3	40.6	28.5	30.0	48.1	58.5	39.9	56.4	65.9	35.4	79.2	74.1	74.3	62.1	66.1	**82.8**
C29→C27	31.2	51.5	31.3	33.6	56.5	64.3	47.1	61.7	74.8	48.2	**85.3**	81.8	78.6	70.7	79.2	84.7
Average	34.8	43.3	35.4	44.8	59.6	63.5	58.8	58.3	73.1	43.4	81.0	80.6	77.1	71.2	68.4	**85.4**

as follows. PACET$_{CDL}$ makes each target sample far away from the source center of the nearest negative class to align two domains. However, due to the uncertainty of the pesudo lables of the target domain, it is likely to cause target samples far away from the same-category source center of the true label, especially in the tasks with many categories. PC$_{G(T\to S)}$ aligns each target sample and its weighted source centroids from a global perspective, reducing the negative impact of the category-wise error in domain alignment.

5) The local cross-domain alignment item PC$_L$ (Eq.(35)) aligns each source sample with its same-category target nearest neighbors. Comparing the performance of PC$_L$ and PC$_{L(T\to S)}$, we can see that the average accuracy of PC$_L$ is 10.6% higher than that of PC$_{L(T\to S)}$. This is because, many target samples may be given wrong pseudo labels, finding the same-category source nearest neighbors for these target samples will increase the error of cross-domain alignment in PC$_{L(T\to S)}$, especially in the tasks with many categories, such as Ar→Cl and C05→C09, the accuracy of PC$_{L(T\to S)}$ decreases seriously. So, we only utilize PC$_L$ to preserve the local spatial relationship across domains.

2) Verification of Different Soft Label Generation Strategies: As the probability information has the contribution to PGCD, we will test which soft label generation strategy is the most

TABLE X
ACCURACIES (%) OF DIFFERENT VARIANTS OF PGCD

Methods	Office-Caltech-10 (SURF) A→D	Office-Caltech-10 ($DeCAF_6$) C→D	Office31 ($DeCAF_7$) W→A	Office-Home (Resnet50) Ar→Cl	CMU-PIE C05→C09	Average
PGCD	**52.2**	**92.4**	**61.2**	**57.7**	**82.8**	**69.3**
GCD	42.7	91.1	58.7	56.7	79.8	65.8
PGD	49.7	90.5	60.6	55.6	65.6	64.4
PGC	46.3	89.8	49.4	56.4	78.6	64.1
$PGCD_S$	52.0	89.8	56.5	56.8	82.2	67.5
$PGCD_T$	47.4	89.5	52.7	57.1	79.2	65.2
PGC_LD	49.7	91.7	61.0	57.3	82.0	68.3
PGC_GD	49.8	91.7	61.1	57.2	77.5	67.5
PD	42.0	90.1	52.4	55.4	41.6	56.3
PD_S	38.2	89.2	49.8	50.2	34.7	52.4
PD_T	41.4	87.3	46.2	47.7	34.3	51.4
PC_{L+G}	45.2	87.3	45.7	56.2	78.1	62.5
PC_L	44.5	85.4	41.9	51.7	76.8	60.1
PC_G	45.1	86.0	45.4	55.9	76.8	61.8
$PC_{G(S\to T)}$	44.9	85.9	44.7	53.6	69.4	59.7
$PC_{G(T\to S)}$	44.6	86.0	45.2	51.1	72.6	59.9
$PACET_{CDL}$	37.6	85.4	40.5	47.7	38.3	49.9
$PC_{L(T\to S)}$	44.3	86.1	41.2	47.7	28.1	49.5

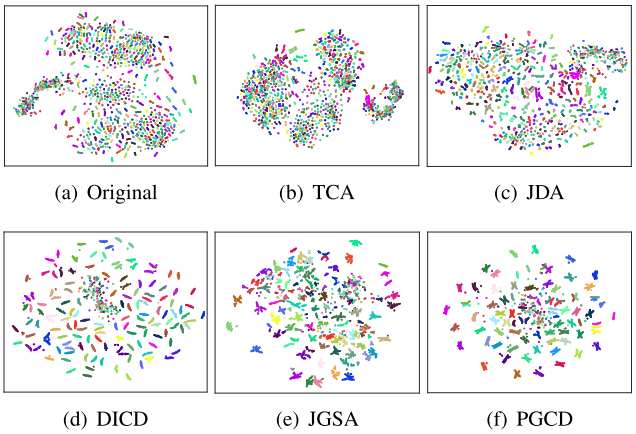
TABLE XI
ACCURACIES (%) OF DIFFERENT SOFT LABEL GENERATION STRATEGIES

Methods	Office-Caltech-10 (SURF) A→D	Office-Caltech-10 ($DeCAF_6$) C→D	Office31 ($DeCAF_7$) W→A	Office-Home (Resnet50) Ar→Cl	CMU-PIE C05→C09	Average
KNN	**52.2**	**92.4**	**61.2**	**57.7**	**82.8**	**69.3**
SVR	47.8	91.7	51.1	52.2	55.3	59.6
RF	42.0	90.5	49.9	50.3	50.9	56.7
SoftMax	46.5	**92.4**	58.1	51.7	55.2	60.8

appropriate one. Comparing with different soft label strategies including SVR, Random Forest (RF), and SoftMax Classifier, the KNN strategy used in PGCD has the best accuracy as shown in Table XI. During the experiment, there are two phenomena that affect the results of the experiment: (1) Soft labels can make the pseudo label more accurate, however, there are too many non-zero values in the soft label vector in SVR, RF, and SoftMax, which will lead to the inaccuracies of the centroids. (2) The number of categories of the cross-domain task also influences the result of the experiments. In methods SVR, RF, and SoftMax, most categories have non-zero probability values, as the categories increase, it will cause irrelevant categories to interfere with the accuracy of centroid construction, which will have a negative impact on the prediction results.

Based on the above findings, we adopt KNN to generate soft labels because we can set the number of neighbors to adjust the number of non-zero and zero elements in the soft label vectors, which can make PGCD have stronger generalization ability especially in datasets with many categories. Further, our model utilizes 1NN as the final model classifier. Utilizing well-matched KNN to generate soft labels can provide a positive effect for the model classifier.

3) Feature Visualization: In order to compare the projection space data distribution of two domains more intuitively, t-SNE visualization is used in this part. PGCD, TCA, JDA, JGSA, and DICD are compared on the task C07→C29 with 68 categories of dataset CMU-PIE. In two domains, samples of each category are represented by the same color, and samples of different categories are represented by the different colors. Through this way, we can observe and verify whether each method can align the two domains and make each category sufficiently discriminative. Observing Fig. 2, TCA and JDA do not enable samples in different domains discriminative enough. DICD encourages the same-category samples compact but does not fully align the same-category clusters. JGSA enables the same-category clusters in different domains closer, but the clusters are not discriminative enough. Compared with these methods, PGCD fully aligns the same-category clusters of two domains, meanwhile, makes same-category samples compact and sufficiently discriminative.



(a) Original (b) TCA (c) JDA

(d) DICD (e) JGSA (f) PGCD

Fig. 2. T-SNE visualizations of PGCD and other compared UDA methods of cross-domain task C07→C29 in CMU-PIE.

4) Parameter Sensitivity: We conduct parameter sensitivity studies with respect to trade-off parameters α, β, and projection space dimension z. To verify the effects of these parameters in PGCD, we study the sensitivity with a wide range of parameter values. The detailed results of α and β in a wide range $[0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]$ are shown in Fig.3. The detailed results of z in a wide range $[15, 20, 30, 50, 70, 100]$ are shown in Fig.4.

As shown in Fig.3, we select a task in each dataset to report the results, while similar charts for all other tasks are not shown due to space limitation. Parameters α and β are not sensitive to datasets Office-Caltech-10 ($DeCAF_6$) and Office-Home. For most tasks, we can get acceptable results when α is under a reasonable range $[0.1, 5]$, meanwhile, β is under a reasonable range $[0.01, 5]$. As shown in Fig.4, z is sensitive to experimental results, because it determines the structure of low-dimension embedding. We set $z = 15$ for Office-Caltech-10 (SURF), $z = 20$ for Office-Caltech-10 ($DeCAF_6$), $z = 30$ for Office31, $z = 70$ for Office-Home, $z = 100$ for CMU-PIE. For each dataset, we use common parameters for all tasks in it.

5) Convergence Analysis: To verify the convergence of the proposed PGCD, we record the change of the objective function value (obj) Eq.(41) and the er value as the number of iterations increases. The $er = |obj(t+1) - obj(t)|$ is the absolute value of the difference between the $t+1$-th objective function value and the t-th objective function value.

As shown in Fig. 5, the objective function value tends to be stable in a limited number of iterations. The objective

TABLE XII
p-VALUE FOR WILCOXON SIGNED-RANK TEST ON OFFICE-HOME

PGCD minus	1NN	SVM	GFK	JDA	JGSA	DICD	DICE	HCA	ACE
	4.88E-04	4.88E-04	4.88E-04	4.88E-04	4.88E-04	1.5E-03	4.88E-04	4.88E-04	4.88E-04
	Shallow methods ⇑				and	**Deep methods ⇓**			
PGCD minus	ResNet	DAN	DANN	JAN	CDAN	TAT	GAACN	GDAN	AEO
	4.88E-04	4.88E-04	4.88E-04	4.88E-04	9.76E-04	2.4E-03	1.22E-02	4.88E-04	0.2334

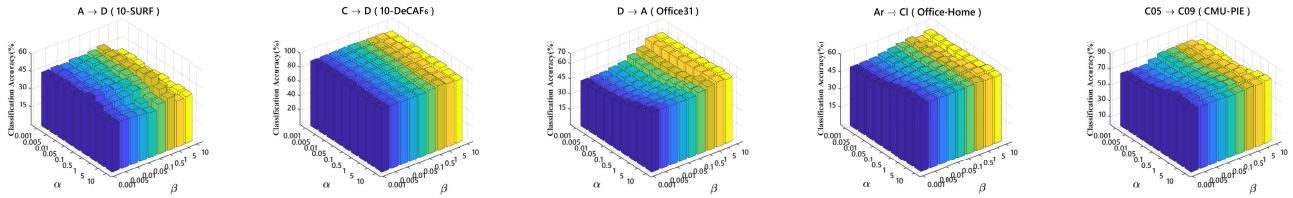


Fig. 3. Classification accuracies of the proposed method with respect to α and β.

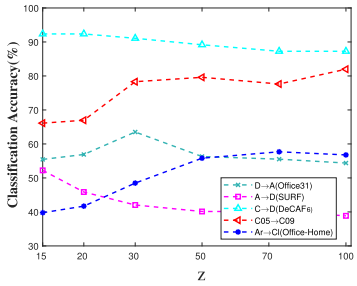


Fig. 4. Classification accuracies of the proposed method with respect to z.

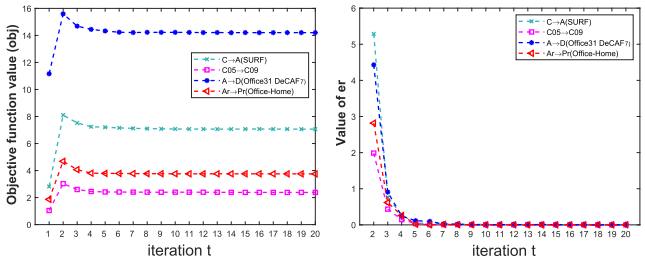


Fig. 5. Convergence curve of PGCD.

value is ascent at the second iteration step on the four cross-domain tasks. The reason is in the first step of iteration, the $\mathbf{L}^G$, $\mathbf{L}^L$, $\mathbf{L}^{I_2}$, and $\sum_{c=1}^{C}\mathbf{M}_c$ are initialized as zero (as shown in Algorithm 1), so the objective value is small in the first iteration. The value of er reduces gradually and tends to zero, which means that the proposed method can satisfy the convergence condition within 20 iterations. Generally speaking, the proposed algorithm PGCD has a good convergence property.

6) Statistical Analysis: In order to further prove the superiority of the proposed PGCD, Wilkinson signed-rank test is used to make statistical analysis of experimental results in this part. The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test that can be used to compare two paired observations. According to the differences and ranks of different methods in each cross-domain task, the p-value under the null hypothesis can be calculated. Office-Home is complex enough with 65 categories, so 12 cross-domain tasks in this dataset are selected for statistical analysis to verify the superiority of PGCD as shown in Table XII. Compare with 9 shallow methods, PGCD has significant differences compared with all methods at 0.05 significant level. Compare with 9 deep methods, PGCD has significant differences compare with 8 methods except for AEO at 0.05 significant level. To sum up, PGCD is significantly superior to most compared methods.

V. CONCLUSION

In this article, we propose a graph embedding framework PGCD for unsupervised domain adaptation. This framework contains a novel cross-domain alignment item to align the local and global geometric structures of two domains from the sample perspective. Meanwhile, we propose a novel class centroid strategy to generate a specific centroid for each target sample to compose the class discriminative transfer feature learning item to make the projected features discriminative enough. In addition, the specific centroids for each proposed item are generated by soft labels and confidence weights, which can reduce the error accumulation of pseudo labels and further enhance the discriminativeness and similarity of two domains. Finally, the proposed items are embedded into the unified graph framework to intuitively describe the relationship between samples and samples during the discriminative transfer feature learning process and domain alignment process in single-domain and cross-domain scenarios.

There are still some promising directions on this work. (1) PGCD only aligns the first-order statistic of the data, how to find the optimal metric is worthy of further study and discussion; (2) Inspired by [46], combining the strategies in our method with the outstanding backbone Transformer to achieve end-to-end deep domain adaptation is worthy of further research.

REFERENCES

[1] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2144–2155, Jun. 2019.
[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[3] J. Liang, R. He, Z. Sun, and T. Tan, "Aggregating randomized clustering-promoting invariant projections for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1027–1042, May 2019.
[4] H. Xu, M. Yang, L. Deng, Y. Qian, and C. Wang, "Neutral cross-entropy loss based unsupervised domain adaptation for semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 4516–4525, 2021.

[5] H. Feng, M. Chen, J. Hu, D. Shen, H. Liu, and D. Cai, "Complementary pseudo labels for unsupervised domain adaptation on person re-identification," *IEEE Trans. Image Process.*, vol. 30, pp. 2898–2907, 2021.

[6] Y. Jiao, H. Yao, and C. Xu, "SAN: Selective alignment network for cross-domain pedestrian detection," *IEEE Trans. Image Process.*, vol. 30, pp. 2155–2167, 2021.

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

[8] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2200–2207.

[9] Y. Chen, S. Song, S. Li, and C. Wu, "A graph embedding framework for maximum mean discrepancy-based domain adaptation algorithms," *IEEE Trans. Image Process.*, vol. 29, pp. 199–213, 2020.

[10] S. Li et al., "Discriminative transfer feature and label consistency for cross-domain image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4842–4856, Nov. 2020.

[11] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4260–4273, Sep. 2018.

[12] L. Yang and P. Zhong, "Robust adaptation regularization based on within-class scatter for domain adaptation," *Neural Netw.*, vol. 124, pp. 60–74, Apr. 2020.

[13] J. Liang, R. He, Z. Sun, and T. Tan, "Exploring uncertainty in pseudo-label guided unsupervised domain adaptation," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106996.

[14] L. Yang and P. Zhong, "Discriminative and informative joint distribution adaptation for unsupervised domain adaptation," *Knowl.-Based Syst.*, vol. 207, Nov. 2020, Art. no. 106394.

[15] Y. Xie, Z. Du, J. Li, M. Jing, E. Chen, and K. Lu, "Joint metric and feature representation learning for unsupervised domain adaptation," *Knowl.-Based Syst.*, vol. 192, Mar. 2020, Art. no. 105222.

[16] L. Tian, Y. Tang, L. Hu, Z. Ren, and W. Zhang, "Domain adaptation by class centroid matching and local manifold self-learning," *IEEE Trans. Image Process.*, vol. 29, pp. 9703–9718, 2020.

[17] M. Jing, J. Zhao, J. Li, L. Zhu, Y. Yang, and H. T. Shen, "Adaptive component embedding for domain adaptation," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3390–3403, Jul. 2021.

[18] B. Tan, Y. Song, E. Zhong, and Q. Yang, "Transitive transfer learning," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1155–1164.

[19] F. Xu, J. Yu, and R. Xia, "Instance-based domain adaptation via multiclustering logistic approximation," *IEEE Intell. Syst.*, vol. 33, no. 1, pp. 78–88, Jan. 2018.

[20] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[21] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[22] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 850–863, Dec. 2016.

[23] P. Razzaghi, P. Razzaghi, and K. Abbasi, "Transfer subspace learning via low-rank and discriminative reconstruction matrix," *Knowl.-Based Syst.*, vol. 163, pp. 174–185, Jan. 2019.

[24] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5150–5158.

[25] Y. Liu, W. Tu, B. Du, L. Zhang, and D. Tao, "Homologous component analysis for domain adaptation," *IEEE Trans. Image Process.*, vol. 29, pp. 1074–1089, 2020.

[26] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 402–410.

[27] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. van den Hengel, "An embarrassingly simple approach to visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3403–3417, Jul. 2018.

[28] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.

[29] W. Wang, H. Wang, Z.-Y. Ran, and R. He, "Learning robust feature transformation for domain adaptation," *Pattern Recognit.*, vol. 114, Jun. 2021, Art. no. 107870.

[30] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[31] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 97–105.

[32] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 2208–2217.

[33] L. Hu, M. Kan, S. Shan, and X. Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4042–4051.

[34] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 1180–1189.

[35] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation, in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2018, pp. 1647–1657.

[36] H. Liu, M. Long, J. Wang, and M. Jordan, "Transferable adversarial training: A general approach to adapting deep classifiers," in *Proc. Int. Conf. Mach. Learn. (LCML)*, vol. 97, Jun. 2019, pp. 4013–4022.

[37] W. Chen and H. Hu, "Generative attention adversarial classification network for unsupervised domain adaptation," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107440.

[38] B. Yang and P. C. Yuen, "Learning adaptive geometry for unsupervised domain adaptation," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107638.

[39] A. Ma, J. Li, K. Lu, L. Zhu, and H. T. Shen, "Adversarial entropy optimization for unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6263–6274, Nov. 2022, doi: 10.1109/TNNLS.2021.3073119.

[40] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, and Q. Tian, "Gradually vanishing bridge for adversarial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12452–12461.

[41] X. Gu, J. Sun, and Z. Xu, "Spherical space domain adaptation with robust pseudo-label loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9098–9107.

[42] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8722–8732.

[43] J. Na, H. Jung, H. J. Chang, and W. Hwang, "FixBi: Bridging domain spaces for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1094–1103.

[44] Z. Yue, Q. Sun, X.-S. Hua, and H. Zhang, "Transporting causal mechanisms for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8579–8588.

[45] S. Li et al., "Domain conditioned adaptation network," in *Proc. Amer. Assoc. Artif. Intell. Conf. (AAAI)*, 2020, pp. 11386–11393.

[46] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, and R. Jin, "CDTrans: Cross-domain transformer for unsupervised domain adaptation," 2021, *arXiv:2109.06165*.

[47] H. Liu, J. Hu, Y. Li, and Z. Wen, "K-means clustering," in *Optimisation: Modeling, Algorithm and Theory*. Beijing, China: Academic, 2020, pp. 106–119.

[48] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2066–2073.

[49] T. Kerdoncuff, R. Emonet, and M. Sebban, "Metric learning in optimal transport for domain adaptation," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2020, pp. 2162–2168.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[52] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, pp. 136–144.

[53] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, Oct. 2016, pp. 443–450.

[54] J. Zhuo, S. Wang, W. Zhang, and Q. Huang, "Deep unsupervised convolutional domain adaptation," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 261–269.

Wenxu Wang received the M.Sc. degree from the Faculty of Information Science and Engineering, Ocean University of China, Shandong, China, in 2019. He is currently pursuing the Ph.D. degree with the College of Information and Electrical Engineering, China Agricultural University, Beijing, China.

His research interests include computer vision, machine learning, and transfer learning.

Zhencai Shen is currently an Associate Professor of applied mathematics with China Agricultural University. He has authored or coauthored more than 40 papers published in leading international journals, including *Science China Mathematics*, *Mediterranean Journal of Mathematics*, *Journal of Algebra*, *Signal Processing*, and *Computers and Electronics in Agriculture*. His research interests include algebra and its applications in applied areas. His current research interests include series of significant aspects of mathematics, including finite group theory, representation theory, fusion systems, and machine learning.

Daoliang Li received the Ph.D. degree in agricultural engineering from China Agricultural University in 1999.

He is currently a Professor with the Department of Electronic Engineering, China Agricultural University, and also the Director of the National Innovation Center for Digital Fishery, China. His research interests include the agricultural Internet of Things, agricultural artificial intelligence, and agricultural information processing.

Ping Zhong is currently a Professor of applied mathematics with China Agricultural University, Beijing, China. She has authored or coauthored over 80 papers published in leading journals, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, *Information Sciences*, *Knowledge-Based Systems*, *Neural Networks*, *Engineering Applications of Artificial Intelligence*, and so on. Her current research interests include machine learning, image processing, and agricultural artificial intelligence.

Yingyi Chen received the Ph.D. degree in agricultural engineering from China Agricultural University in 2008.

He is currently a Professor with the Department of Computer Engineering, China Agricultural University. His research interests include the agricultural Internet of Things, agricultural artificial intelligence, and agricultural information processing.