

Domain Adaptation by Class Centroid Matching and Local Manifold Self-Learning

Lei Tian*, Yongqiang Tang*, Liangchen Hu, Zhida Ren, and Wensheng Zhang

Abstract—Domain adaptation has been a fundamental technology for transferring knowledge from a source domain to a target domain. The key issue of domain adaptation is how to reduce the distribution discrepancy between two domains in a proper way such that they can be treated indifferently for learning. In this paper, we propose a novel domain adaptation approach, which can thoroughly explore the data distribution structure of target domain. Specifically, we regard the samples within the same cluster in target domain as a whole rather than individuals and assigns pseudo-labels to the target cluster by class centroid matching. Besides, to exploit the manifold structure information of target data more thoroughly, we further introduce a local manifold self-learning strategy into our proposal to adaptively capture the inherent local connectivity of target samples. An efficient iterative optimization algorithm is designed to solve the objective function of our proposal with theoretical convergence guarantee. In addition to unsupervised domain adaptation, we further extend our method to the semi-supervised scenario including both homogeneous and heterogeneous settings in a direct but elegant way. Extensive experiments on seven benchmark datasets validate the significant superiority of our proposal in both unsupervised and semi-supervised manners.

Index Terms—domain adaptation, class centroid matching, local manifold self-learning.

I. INTRODUCTION

IN many real-world applications, data are generally collected under different conditions, thus hardly satisfying the identical probability distribution hypothesis which is known as a foundation of statistical learning theory. This situation naturally leads to a crucial issue that a classifier trained on a well-annotated source domain cannot be applied to a related but different target domain directly. To surmount this issue, as an important branch of transfer learning, considerable efforts have been devoted to domain adaptation [1]. By far, domain adaptation has been a fundamental technology for cross-domain knowledge discovery, and been considered in various tasks, such as object recognition [2], [3], face recognition [4], [5] and person re-identification [6].

The major issue for domain adaptation is how to reduce the distributions difference between the source and target

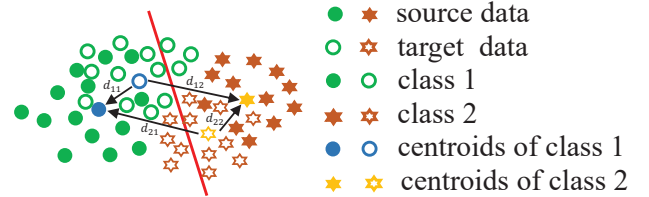


Figure 1. A toy example of misclassification caused by the ignorance of structure information of data distribution, where d_{ij} indicates the distance between the centroids of i -th class in target domain and the centroids of j -th class in source domain.

domains [7]. Most of recent works aim to seek a common feature space where the distribution difference across domains are minimized [8], [9], [10], [11], [12]. To accomplish this, various metrics have been proposed to measure the discrepancy of data distribution, such as correlation distances [13] or maximum mean discrepancy (MMD) [14]. In this work, we try to thoroughly explore the data distribution structure of target domain with a class centroid matching scheme. To illustrate our motivation more explicitly, a toy example is shown in Fig. 1. The red line is a discriminant hyperplane trained on source data in the projected feature space. As we can see, when the distribution discrepancy between two domains still exists, the hyperplane tends to misclassify target samples if they are labeled independently. In such case, the misclassified samples will seriously mislead the learning of the common feature space in the subsequent iterations, and ultimately cause significant performance drop. Nevertheless, by the in-depth analysis of Fig. 1, we could further discover that the class centroids in target domain can be well matched to their corresponding class centroids in source domain. In light of this, when we treat the samples within the same cluster in target domain as a whole rather than individuals, they could share identical label and obtain accurate classification result. Motivated by this insight, in this paper, instead of labeling target samples individually, we aim to introduce a novel domain adaptation approach that assigns pseudo-labels to target samples with the guidance of class centroids in two domains, such that the data distribution structure of target domain can be emphasized.

To achieve this goal, the first issue to be handled is the calculation of class centroids in two domains. Due to the labels available in source domain, we can conveniently acquire the class centroids of source data in the common feature space by calculating the mean of samples in the same class. For target domain where the labels are absent, we resort to the classical K -means clustering algorithm [15] which has been

* indicates equal contributions.

L. Tian, Z. Ren and W. Zhang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China, and University of Chinese Academy of Sciences, Beijing, 101408, China. E-mail: {tianlei2017, renzhida2017}@ia.ac.cn, zhang-wenshengia@hotmail.com.

Y. Tang is with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. E-mail: tangyongqiang2014@ia.ac.cn

L. Hu is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China. E-mail: hlc_clear@foxmail.com.

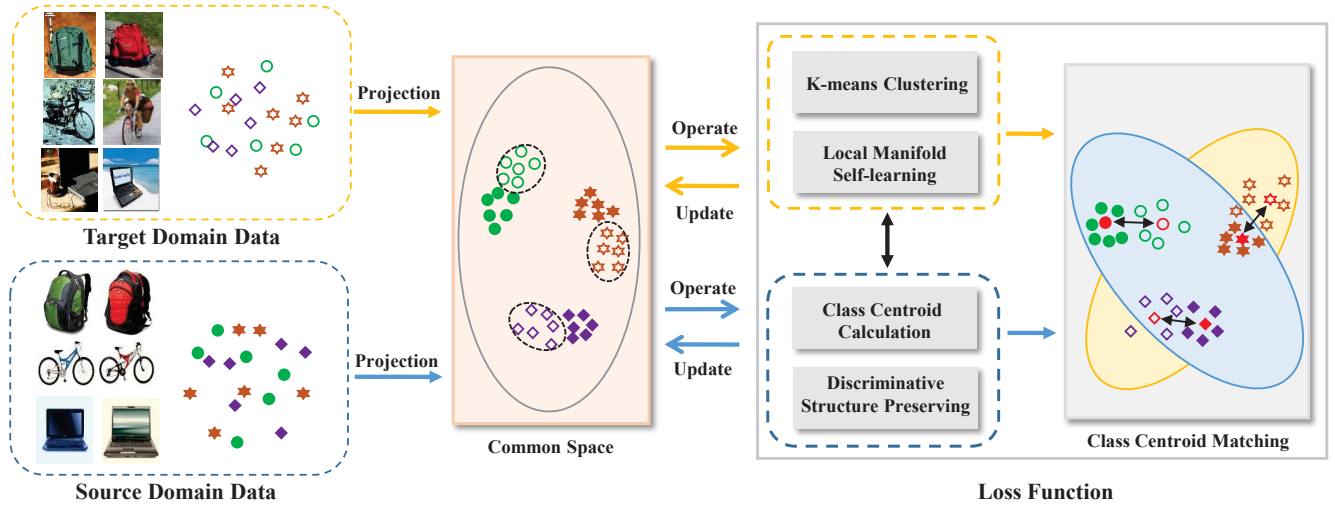


Figure 2. Flowchart of our proposed method. We reformulate the domain adaptation problem as the class centroid matching problem, such that the structure information of data distribution can be exploited. For target data, we further introduce the local manifold self-learning strategy to explore the inherent local connectivity structure. The yellow and blue pipelines indicate the operations for target domain and source domain, respectively.

widely used to partition unlabeled data into several groups where similar samples in the same group can be represented by a specific cluster prototype. Intuitively, the cluster prototypes obtained by K -means algorithm can be regarded as a good approximation for the class centroids of target domain. After obtaining the class centroids of both source data and target data, the distribution discrepancy minimization problem in domain adaptation can be reformulated as the class centroid matching problem which can be solved efficiently by the nearest neighbor search.

The next key concern is how to improve the quality of class centroids, which is crucial to the performance of our approach. In this paper, for labeled source data, we adopt an intuitive strategy that encourages the samples in the identical class to be as close as possible in the projected space, such that the discriminative structure information of source domain can be preserved [16]. As for unlabeled target data, it has been shown that the clustering performance can be significantly enhanced if the local manifold structure is exploited [17], [18]. Nevertheless, most of existing manifold learning methods highly depend on the predefined similarity matrix built in the original feature space [19], [20], and thus may fail to capture the inherent local structure of high-dimensional data due to the curse of dimensionality. To tackle this problem, inspired by the recently proposed adaptive neighbors learning method [21], we introduce a local structure self-learning strategy into our proposal. Specifically, we learn the data similarity matrix according to the local connectivity in the projected low-dimensional feature space rather than the original high-dimensional space, such that the intrinsic local manifold structure of target data can be captured adaptively.

Based on above analysis, a novel domain adaptation method, which can adequately exploit the data distribution structure of target domain by jointly class Centroid Matching and local Manifold Self-learning (CMMS), is naturally proposed. It is noteworthy that, more recently, the need for tackling semi-supervised domain adaptation (SDA) problem is growing as

there may be some labeled target samples in practice [22], [23], [24], [25], [26]. While unsupervised domain adaptation (UDA) methods are well established, most of them cannot be naturally applied to the semi-supervised scenario. Excitingly, the proposed CMMS can be extended to SDA including both homogeneous and heterogeneous settings in a direct but elegant way. The flowchart of our proposed CMMS is shown in Fig. 2. The main contributions of this paper are summarized as follows:

- We propose a novel domain adaptation method called CMMS, which can thoroughly explore the data distribution structure of target domain via jointly class centroid matching and local manifold self-learning.
- We present an efficient optimization algorithm to solve the objective function of the proposal, with theoretical convergence guarantee.
- In addition to unsupervised domain adaptation, we further extend our approach to the semi-supervised scenario including both homogeneous and heterogeneous settings.
- We conduct extensive evaluation of our method on seven benchmark datasets, which validates the superior performance of our method in both unsupervised and semi-supervised manners.

The rest of this paper is organized as follows. Section II previews some related literature. Section III shows our proposal, the optimization algorithm, the convergence and complexity analysis. We describe our semi-supervised extension in Section IV. Massive experimental results are shown in Section V. Finally, we conclude this paper in Section VI.

II. RELATED WORK

In this section, we review some previous works closely related to this paper. First, we briefly review the unsupervised domain adaptation methods. Next, related studies of semi-supervised domain adaptation are reviewed. Finally, we introduce some local manifold learning techniques.

A. Unsupervised Domain Adaptation

Unsupervised domain adaptation aims to handle the scenario where labeled samples are only available from the source domain and there exists different distributions between source and target domains. In the past decades, numerous methods have been proposed to overcome the distribution discrepancy.

Existing UDA methods can be classified as: 1) instance reweighting [27], [28], 2) classifier adaptation [7], [29], and 3) feature adaptation [8], [9], [10], [11]. We refer the interested readers to [30], which contains an excellent survey. Our proposal falls into the third category, i.e., feature adaptation, which addresses domain shift by either searching intermediate subspaces to achieve domain transfer [10], [11] or learning a common feature space where the source and target domains have similar distributions [8], [9]. In this paper, we focus on the latter line. Among existing works, TCA [8] is a pioneering approach, which learns a transformation matrix to align marginal distribution between two domains via MMD. Later, JDA [9] considers conditional distribution alignment by forcing the class means to be close to each other. In the subsequent research, several works further propose to employ the discriminative information to facilitate classification performance. For instance, Li *et al.* [16] utilized the discriminative information for the source and target domains by encouraging intra-class compactness and inter-class dispersion. Liang *et al.* [31] achieved this goal by promoting class clustering.

Despite the promising performance, all the above methods classify target samples independently, which may cause misclassification since the structure information of data distribution in target domain is ignored. To tackle this issue, several recent works attempt to exploit the distribution structure of target data via clustering. For example, Liang *et al.* [32] proposed to seek a subspace where the target centroids are forced to approach those in the source domain. Inspired by the fact that the target samples are well clustered in the deep feature space, Wang *et al.* [33] proposed a selective pseudo-labeling approach (SPL) based on structured prediction, which is the most relevant to our proposal. Nevertheless, our proposal is significantly different from it. First, these two approaches are essentially two different formalization frameworks. To be specific, our proposal integrates the projection matrix learning, the class centroid calculation and the class centroid matching into a unified optimization objective, which is more intuitive and concise. Second, in our framework, we further present the specific strategies for source and target data respectively, including discriminative structure preserving for source data and local manifold self-learning for target data, to improve the quality of class centroids. Third, our formalization owns better scalability, which can be naturally extended to semi-supervised scenario with little modification of optimization algorithm.

More recently, deep domain adaptation methods has attracted much more attention in the past few years [30], which are generally achieved by adding adaptation layers to align the means of distributions [34], [35], [41] or adding a subnetwork to confuse the source and target domains [36], [37], [38], [39], [40]. For instance, Pinheiro [39] compared the embedding of a target image with the source prototypes and the label

that best matches it is given. Chen *et al.* [40] employed a different manner by matching the source and target prototypes for each class via adaptive prototype alignment. Pan *et al.* [41] innovatively proposed transferrable prototypical networks (TPN) for unsupervised domain adaptation, which enforces the prototypes of each class in source and target domains to be close in the embedding space and the score distributions predicted by prototypes separately on source and target data to be similar. Both TPN and our CMMS share the similar idea of using prototypes to represent the class distribution. Nevertheless, there are significant differences between the two approaches. First, in the process of assigning pseudo-label to target data, TPN treats the target samples as individuals and independently matches each target sample to the nearest prototype in the source domain. As shown in the toy example in Fig.1, the drawback of such procedure lies in the neglect of the distribution structure of target data. We handle this issue by first treating the samples within the same cluster in target domain as a whole, and then assigning the label of nearest source class to the target cluster. Second, the exploration of local manifold structure of target data plays a vital role in the learning of common space, which is ignored by TPN. By contrast, we introduce local manifold self-learning strategy into our CMMS to capture the intrinsic local connectivity.

B. Semi-supervised Domain Adaptation

Unlike the unsupervised domain adaptation that no labels are available, in practice, a more common scenario is that the target domain contains a few labeled samples. Such scenario leads to a promising research direction, which is referred as the semi-supervised domain adaptation.

According to the property of sample features, SDA algorithms are developed in two different settings: 1) homogeneous setting, i.e., the source and target data are sampled from the same feature space; 2) heterogeneous setting, i.e., the source and target data often have different feature dimensions. In the homogeneous setting, the labeled target samples are used in various ways. For example, Hoffman *et al.* [22] jointly learnt the transformation matrix and classifier parameters, forcing the source and target samples with identical label have high similarity. Similarly, Herath *et al.* [23] proposed to learn the structure of a Hilbert space to reduce the dissimilarity between labeled samples and further match the source and target domains via the second order statistics. Recently, based on Fredholm integral, Wang *et al.* [24] proposed to learn a cross-domain kernel classifier that can classify the labeled target data correctly using square loss function or hinge loss function. In the heterogeneous setting, relieving feature discrepancy and reducing distribution divergence are two inevitable issues [42]. For the first issue, one incredibly simple approach [25] is to use the original features or zeros to augment each transformed sample into same size. Another natural approach [26] is to learn two projection matrices to derive a domain-invariant feature subspace, one for each domain. Recently, after employing two matrices to project the source and target data to a common feature space, Li *et al.* [42] employed a shared codebook to match the new feature representations on the same bases.

For the second issue, one favorite solution is to minimize the MMD distance of the source and target domains [26], [42]. Additionally, Tsai *et al.* [43] proposed a representative landmark selection approach, which is similar to instance reweighting in the UDA scenario. When we obtain a limited amount of labeled target samples, manifold regularization, an effective strategy for semi-supervised learning, has also been employed by several previous works [44], [45].

In contrast to these SDA methods, our semi-supervised extension is quite simple and intuitive. To be specific, the labeled target data are used to improve the cluster prototypes learning of the unlabeled target data. Besides, connections between the labeled and unlabeled target data are built, which is a common strategy to develop a semi-supervised model. Different from the homogeneous setting where a unified projection is learnt, we learn two projection matrices in the heterogeneous setting like [26]. Notably, the resulting optimization problems in two settings own the same standard formula and can be solved by the same algorithm in UDA scenario with just very tiny modifications.

C. Local Manifold Learning

The goal of local manifold learning is to capture the underlying local manifold structure of the given data in the original high-dimensional space and preserve it in the low-dimensional embedding. Generally, local manifold learning methods contain three main steps: 1) selecting neighbors; 2) computing affinity matrix; 3) calculating the low-dimensional embedding [46].

Local linear embedding [19] and Laplacian eigenmaps [20] are two typical methods. In local linear embedding, the local manifold structure is captured by linearly reconstructing each sample using the corresponding neighbors in the original space and the reconstruction coefficients are preserved in the low-dimensional space. In Laplacian eigenmaps, the adjacency matrix of given data is obtained in the original feature space using Gaussian function. However, the local manifold structure is artificially captured using pairwise distances with heat kernel, which brings relatively weak representation for the ignorance of the properties of local neighbors [46]. Recently, to learn a more reliable adjacency matrix, Nie *et al.* [21] proposed to assign the neighbors of each sample adaptively based on the Euclidean distances in the low-dimensional space. This strategy has been widely utilized in clustering [47], feature selection [48] and feature representation learning [49].

In domain adaptation problems, several works have borrowed the advantages of local manifold learning. For example, Long *et al.* [7] and Wang *et al.* [29] employed manifold regularization to maintain the manifold consistency underlying the marginal distributions of two domains. Hou *et al.* [50] and Li *et al.* [51] used label propagation to predict target labels. However, they all calculate adjacency matrix in the original high-dimensional space with the predefined distance measurement, which is unreliable due to the curse of dimensionality. By contrast, our proposal can capture and employ the inherent local manifold structure of target data adaptively, thus leading to superior performance.

TABLE I
FREQUENTLY USED NOTATIONS AND DESCRIPTIONS

| Notation | Description |
|---|--|
| $\mathbf{X}_s/\mathbf{X}_t$ | source/target original data |
| n_s/n_t | number of source/target samples |
| \mathbf{P} | projection matrix |
| \mathbf{F} | target cluster centroids |
| \mathbf{G}_t | target label matrix |
| \mathbf{S} | target adjacency matrix |
| \mathbf{H} | centering matrix |
| \mathbf{I}_d | identity matrix with dimension d |
| m | dimension of original features |
| d | dimension of projected features |
| C | number of shared class |
| n_s^c | number of source samples in class c |
| $\mathbf{0}_{p \times q}/\mathbf{1}_{p \times q}$ | a matrix of size $p \times q$ with all elements as 0/1 |
| $\mathbf{0}_p/\mathbf{1}_p$ | a column vector of size p with all elements as 0/1 |

III. PROPOSED METHOD

In this section, we first introduce the notations and basic concepts used throughout this paper. Then, the details of our approach are described. Next, an efficient algorithm is designed to solve the optimization problem of our proposal. Finally, the convergence and complexity analysis of the optimization algorithm are given.

A. Notations

A domain \mathcal{D} contains a feature space χ and a marginal probability distribution $P(\mathbf{X})$, where $\mathbf{X} \in \chi$. For a specific domain, a task \mathcal{T} consists of a label space \mathcal{Y} and a labeling function $f(\mathbf{x})$, denoted by $\mathcal{T} = \{\mathcal{Y}, f(\mathbf{x})\}$ [1]. For simplicity, we use subscripts s and t to describe the source domain and target domain, respectively.

We denote the source domain data as $\mathcal{D}_s = \{\mathbf{X}_s, \mathbf{Y}_s\} = \{(\mathbf{x}_{si}, y_{si})\}_{i=1}^{n_s}$, where $\mathbf{x}_{si} \in \mathbb{R}^m$ is a source sample and $y_{si} \in \mathbb{R}$ is the corresponding label. Similarly, we denote the target domain data as $\mathcal{D}_t = \{\mathbf{X}_t\} = \{\mathbf{x}_{tj}\}_{j=1}^{n_t}$, where $\mathbf{x}_{tj} \in \mathbb{R}^m$. For clarity, we show the key notations used in this paper and the corresponding descriptions in Table I.

B. Problem Formulation

The core idea of our CMMS lies in the emphasis on data distribution structure by class centroid matching of two domains and local manifold structure self-learning for target data. The overall framework of CMMS can be stated by the following formula:

$$\min_{\mathbf{P}, \mathbf{F}, \mathbf{G}_t, \mathbf{S}} \Omega(\mathbf{P}, \mathbf{F}) + \alpha \Theta(\mathbf{P}, \mathbf{F}, \mathbf{G}_t) + \gamma \Psi(\mathbf{P}, \mathbf{S}) + \beta \Phi(\mathbf{P}) \quad (1)$$

The first term $\Omega(\mathbf{P}, \mathbf{F})$ is used to match class centroids. $\Theta(\mathbf{P}, \mathbf{F}, \mathbf{G}_t)$ is the clustering term for target data in the projected space. $\Psi(\mathbf{P}, \mathbf{S})$ is employed to capture the data structure information. $\Phi(\mathbf{P})$ is the regularization term to avoid overfitting. Hyper-parameters α , β and γ are employed to balance the influence of different terms. Next, we will introduce these items in detail.

1) *K-means Clustering for Target Data*: In our CMMS, we borrow the idea of clustering to obtain the cluster prototypes which can be regarded as the pseudo class centroids. In such case, the sample distribution structure information of target data can be acquired. To achieve this goal, various existing clustering algorithms can be our candidates. Without loss of generality, for the sake of simplicity, we adopt the classical *K*-means algorithm to get the cluster prototypes in this paper. Thus, we have the following formula:

$$\Theta(\mathbf{P}, \mathbf{F}, \mathbf{G}_t) = \|\mathbf{P}^T \mathbf{X}_t - \mathbf{F} \mathbf{G}_t^T\|_F^2 \quad (2)$$

where $\mathbf{P} \in \mathbb{R}^{m \times d}$ is the projection matrix, $\mathbf{F} \in \mathbb{R}^{d \times C}$ is the cluster centroids of target data, $\mathbf{G}_t \in \mathbb{R}^{n_t \times C}$ is the cluster indicator matrix of target data which is defined as $(\mathbf{G}_t)_{ij} = 1$ if the cluster label of \mathbf{x}_{ti} is j , and $(\mathbf{G}_t)_{ij} = 0$ otherwise.

2) *Class Centroid Calculation for Source Data and Class Centroid Matching of Two Domains*: Once the cluster prototypes of target data are obtained, we can reformulate the distribution discrepancy minimization problem in domain adaptation as the class centroid matching problem. Note that the class centroids of source data can be obtained exactly by calculating the mean value of sample features in the identical class. In this paper, we solve the class centroid matching problem by the nearest neighbor search since it is simple and efficient. Specifically, we search the nearest source class centroid for each target cluster centroid, and minimize the sum of distance of each pair of class centroids. Finally, the class centroid matching of two domains is formulated as:

$$\Omega(\mathbf{P}, \mathbf{F}) = \|\mathbf{P}^T \mathbf{X}_s \mathbf{E}_s - \mathbf{F}\|_F^2 \quad (3)$$

where $\mathbf{E}_s \in \mathbb{R}^{n_s \times C}$ is a constant matrix used to calculate the class centroids of source data in the projected space with each element $\mathbf{E}_{ij} = 1/n_s^j$ if $y_{si} = j$, and $\mathbf{E}_{ij} = 0$ otherwise.

3) *Local Manifold Self-Learning for Target Data*: In our proposed CMMS, the cluster prototypes of target samples are actually the approximation of their corresponding class centroids. Hence, the quality of cluster prototypes plays an important role in the final performance of our CMMS. Existing works have proven that the performance of clustering can be significantly improved by the exploiting of local manifold structure. Nevertheless, most of them highly depend on the predefined adjacent matrix in the original feature space, and thus fail to capture the inherent local manifold structure of high-dimensional data due to the curse of dimensionality. For this issue, inspired by the recent work [21], we propose to introduce a local manifold self-learning strategy into our CMMS. Instead of predefining the adjacent matrix in the original high-dimensional space, we adaptively learn the data similarity according to the local connectivity in the projected low-dimensional space, such that the intrinsic local manifold structure of target data can be captured. The formula of local manifold self-learning is shown as follows:

$$\begin{aligned} \Psi_t(\mathbf{P}, \mathbf{S}) &= \sum_{i,j=1}^{n_t} \|\mathbf{P}^T \mathbf{x}_{ti} - \mathbf{P}^T \mathbf{x}_{tj}\|_2^2 S_{ij} + \delta S_{ij}^2 \\ &= 2\text{tr}(\mathbf{P}^T \mathbf{X}_t \mathbf{L}_t \mathbf{X}_t^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2 \\ \text{s.t. } \mathbf{S} \mathbf{1}_{n_t} &= \mathbf{1}_{n_t}, 0 \leq S_{ij} \leq 1 \end{aligned} \quad (4)$$

where $\mathbf{S} \in \mathbb{R}^{n_t \times n_t}$ is the adjacency matrix in target domain and δ is a hyper-parameter. \mathbf{L}_t is the corresponding graph laplacian matrix calculated by $\mathbf{L}_t = \mathbf{D} - \mathbf{S}$, where \mathbf{D} is a diagonal matrix with each element $\mathbf{D}_{ii} = \sum_{j \neq i} \mathbf{S}_{ij}$.

4) *Discriminative Structure Preserving for Source Data*: The above descriptions have highlighted the main components of our CMMS. Intuitively, a reasonable hypothesis for source data is that the samples in the identical class should be as close as possible in the projected space, such that the discriminative structure information of source domain can be preserved. As one trivial but effective trick, inspired by [16], we formulate this thought as follows:

$$\begin{aligned} \Psi_s(\mathbf{P}) &= \sum_{c=1}^C \frac{1}{n_s^c} \sum_{y_{si}, y_{sj}=c} \|\mathbf{P}^T \mathbf{x}_{si} - \mathbf{P}^T \mathbf{x}_{sj}\|_2^2 \\ &= \sum_{i,j=1}^{n_s} \|\mathbf{P}^T \mathbf{x}_{si} - \mathbf{P}^T \mathbf{x}_{sj}\|_2^2 W_{ij} = 2\text{tr}(\mathbf{P}^T \mathbf{X}_s \mathbf{L}_s \mathbf{X}_s^T \mathbf{P}) \end{aligned} \quad (5)$$

where $\text{tr}(\cdot)$ is the trace operator and \mathbf{W} is the similarity matrix for source data, which is defined as $W_{ij} = 1/n_s^c$ if $y_{si} = y_{sj} = c$, and 0, otherwise. The coefficient $1/n_s^c$ is used to remove the effects of different class sizes [52]. \mathbf{L}_s is the respective laplacian matrix and is defined as:

$$(\mathbf{L}_s)_{ij} = \begin{cases} 1 - \frac{1}{n_s^c}, & \text{if } i = j; \\ -\frac{1}{n_s^c}, & \text{if } i \neq j, y_{si} = y_{sj} = c; \\ 0, & \text{otherwise.} \end{cases}$$

For simplicity, we denote $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t]$ and $\mathbf{L} = \text{diag}(2\mathbf{L}_s, 2\mathbf{L}_t)$. By combining Eq.(4) and Eq.(5), we obtain a general $\Psi(\mathbf{P}, \mathbf{S})$ term which can capture the diverse structure information of both source and target data:

$$\begin{aligned} \Psi(\mathbf{P}, \mathbf{S}) &= 2\text{tr}(\mathbf{P}^T \mathbf{X}_s \mathbf{L}_s \mathbf{X}_s^T \mathbf{P}) + 2\text{tr}(\mathbf{P}^T \mathbf{X}_t \mathbf{L}_t \mathbf{X}_t^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2 \\ &= \text{tr}(\mathbf{P}^T [\mathbf{X}_s, \mathbf{X}_t] \begin{bmatrix} 2\mathbf{L}_s & \mathbf{0}_{n_s \times n_t} \\ \mathbf{0}_{n_t \times n_s} & 2\mathbf{L}_t \end{bmatrix} [\mathbf{X}_s, \mathbf{X}_t]^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2 \\ &= \text{tr}(\mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2 \\ \text{s.t. } \mathbf{S} \mathbf{1}_{n_t} &= \mathbf{1}_{n_t}, \quad 0 \leq S_{ij} \leq 1 \end{aligned} \quad (6)$$

Besides, to avoid overfitting and improve the generalization capacity, we further add an *F*-norm regularization term to the projection matrix \mathbf{P} :

$$\Phi(\mathbf{P}) = \|\mathbf{P}\|_F^2 \quad (7)$$

So far, by combining Eq.(2), (3), (6) and (7), we arrive at our final CMMS formulation:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{F}, \mathbf{G}_t, \mathbf{S}} & \|\mathbf{P}^T \mathbf{X}_s \mathbf{E}_s - \mathbf{F}\|_F^2 + \alpha \|\mathbf{P}^T \mathbf{X}_t - \mathbf{F} \mathbf{G}_t^T\|_F^2 \\ & + \beta \|\mathbf{P}\|_F^2 + \gamma (\text{tr}(\mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2) \\ \text{s.t. } & \mathbf{P}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{P} = \mathbf{I}_d, \quad \mathbf{G}_t \in \{0, 1\}^{n_t \times C}, \\ & \mathbf{S} \mathbf{1}_{n_t} = \mathbf{1}_{n_t}, \quad 0 \leq S_{ij} \leq 1 \end{aligned} \quad (8)$$

where \mathbf{I}_d is an identity matrix of dimension d and \mathbf{H} is centering matrix defined as $\mathbf{H} = \mathbf{I}_{n_s+n_t} - \frac{1}{n_s+n_t} \mathbf{1}_{(n_s+n_t) \times (n_s+n_t)}$. The first constraint in (8) is inspired by principal component analysis, which aims to maximize projected data variance [9].

For the sake of simplified format, we reformulate the objective function in (8) as the following standard formula:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{F}, \mathbf{G}, \mathbf{S}} \quad & \|\mathbf{P}^T \mathbf{X} \mathbf{E} - \mathbf{F}\|_F^2 + \alpha \|\mathbf{P}^T \mathbf{X} \mathbf{V} - \mathbf{F} \mathbf{G}^T\|_F^2 \\ & + \beta \|\mathbf{P}\|_F^2 + \gamma (\text{tr}(\mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2) \quad (9) \\ \text{s.t.} \quad & \mathbf{P}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{P} = \mathbf{I}_d, \quad \mathbf{G} \in \{0, 1\}^{n \times C}, \\ & \mathbf{S} \mathbf{1}_{n_t} = \mathbf{1}_{n_t}, \quad 0 \leq S_{ij} \leq 1 \end{aligned}$$

where $n = n_s + n_t$, $\mathbf{V} = \text{diag}(\mathbf{0}_{n_s \times n_s}, \mathbf{I}_{n_t})$, $\mathbf{E} = [\mathbf{E}_s; \mathbf{0}_{n_t \times C}]$, $\mathbf{G} = [\mathbf{0}_{n_s \times C}; \mathbf{G}_t]$ and $\mathbf{L} = \text{diag}(\mathbf{L}_s, \mathbf{L}_t)$.

C. Optimization Procedure

According to the objective function of our CMMS in Eq.(9), there are four variables \mathbf{F} , \mathbf{P} , \mathbf{G} , \mathbf{S} that need to be optimized. Since it is not jointly convex for all variables, we update each of them alternatively while keeping the other variables fixed. Specifically, each subproblem is solved as follows:

1. F-subproblem: When \mathbf{P} , \mathbf{G} and \mathbf{S} are fixed, the optimization problem (9) becomes:

$$\min_{\mathbf{F}} \|\mathbf{P}^T \mathbf{X} \mathbf{E} - \mathbf{F}\|_F^2 + \alpha \|\mathbf{P}^T \mathbf{X} \mathbf{V} - \mathbf{F} \mathbf{G}^T\|_F^2 \quad (10)$$

By setting the derivative of (10) with respect to \mathbf{F} as 0, we obtain:

$$\mathbf{F} = (\mathbf{P}^T \mathbf{X} \mathbf{E} + \alpha \mathbf{P}^T \mathbf{X} \mathbf{V} \mathbf{G}) (\alpha \mathbf{G}^T \mathbf{G} + \mathbf{I}_C)^{-1} \quad (11)$$

2. P-subproblem: Substituting Eq.(11) into Eq.(9) to replace \mathbf{F} , we can get the following subproblem:

$$\begin{aligned} \min_{\mathbf{P}} \quad & \text{tr}(\mathbf{P}^T (\mathbf{X} \mathbf{R} \mathbf{X}^T + \gamma \mathbf{X} \mathbf{L} \mathbf{X}^T + \beta \mathbf{I}_m) \mathbf{P}) \\ \text{s.t.} \quad & \mathbf{P}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{P} = \mathbf{I}_d \end{aligned} \quad (12)$$

where $\mathbf{R} = \mathbf{E} \mathbf{E}^T - \mathbf{E} (\alpha \mathbf{G}^T \mathbf{G} + \mathbf{I}_C)^{-1} (\mathbf{E} + \alpha \mathbf{V} \mathbf{G})^T + \alpha \mathbf{V} \mathbf{V}^T - \alpha \mathbf{V} \mathbf{G} (\alpha \mathbf{G}^T \mathbf{G} + \mathbf{I}_C)^{-1} (\mathbf{E} + \alpha \mathbf{V} \mathbf{G})^T$. The above problem can be transformed to a generalized eigenvalue problem as follows:

$$(\mathbf{X} \mathbf{R} \mathbf{X}^T + \gamma \mathbf{X} \mathbf{L} \mathbf{X}^T + \beta \mathbf{I}_m) \mathbf{P} = \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{P} \mathbf{\Pi} \quad (13)$$

where $\mathbf{\Pi} = \text{diag}(\pi_1, \pi_2, \dots, \pi_d) \in \mathbb{R}^{d \times d}$ is a diagonal matrix with each element as a Lagrange Multiplier. Then the optimal solution is obtained by calculating the eigenvectors of Eq.(13) corresponding to the d -smallest eigenvalues.

3. G-subproblem: In variable \mathbf{G} , only \mathbf{G}_t needs to be updated. With \mathbf{P} , \mathbf{F} and \mathbf{S} fixed, the optimization problem with regard to \mathbf{G}_t is equal to minimizing Eq.(2). Like K -means clustering, we can solve it by assigning the label of each target sample to its nearest cluster centroid. To this end, we have:

$$(\mathbf{G}_t)_{ik} = \begin{cases} 1, & \text{if } k = \arg \min_j \|\mathbf{P}^T \mathbf{x}_{ti} - \mathbf{F}(:, j)\|_2^2 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

4. S-subproblem: When \mathbf{G} , \mathbf{F} and \mathbf{P} are fixed, the optimization problem with regard to \mathbf{S} is equal to minimizing Eq.(5). Actually, we can divide it into n_t independent subproblems with each formulated as:

$$\min_{\mathbf{S}_{i,:}, \mathbf{1}_{n_t}} \sum_{j=1}^{n_t} \|\mathbf{P}^T \mathbf{x}_{ti} - \mathbf{P}^T \mathbf{x}_{tj}\|_2^2 S_{ij} + \delta S_{ij}^2 \quad (15)$$

Algorithm 1: CMMS for UDA

Input: Source data $\{\mathbf{X}_s, \mathbf{Y}_s\}$; target data $\{\mathbf{X}_t\}$; initial target label matrix \mathbf{G}_t ; initial adjacency matrix \mathbf{S} ; hyper-parameters $\gamma = 5.0$, α , β ; subspace dimensionality $d = 100$; neighborhood size $k = 10$; maximum iteration $T = 10$.

Output: Target label matrix \mathbf{G}_t

```

1  $t = 0$ ;
2 while not converge and  $t \leq T$  do
3   // Projection matrix  $\mathbf{P}$ 
4   Update  $\mathbf{P}$  by solving the generalized eigenvalue
   problem in (13);
5   // Cluster prototype matrix of target data  $\mathbf{F}$ 
6   Update  $\mathbf{F}$  by (11);
7   // Label assignment matrix of target data  $\mathbf{G}_t$ 
8   Update each row of  $\mathbf{G}_t$  by (14);
9   // Local adjacency matrix of target data  $\mathbf{S}$ 
10  Update each row of  $\mathbf{S}$  by solving (16);
11   $t = t + 1$ ;
12 end
13 Return Target label matrix  $\mathbf{G}_t$ .
```

where $\mathbf{S}_{i,:}$ is the i -th row of \mathbf{S} . By defining $A_{ij} = \|\mathbf{P}^T \mathbf{x}_{ti} - \mathbf{P}^T \mathbf{x}_{tj}\|_2^2$, the above problem can be written as:

$$\min_{\mathbf{S}_{i,:}, \mathbf{1}_{n_t}} \sum_{j=1}^{n_t} \|\mathbf{S}_{i,:}\|_2^2 + \frac{A_{ij}}{2\delta} \quad (16)$$

The corresponding Lagrangian function is:

$$\min_{\mathbf{S}_{i,:}} \|\mathbf{S}_{i,:}\|_2^2 + \frac{A_{ij}}{2\delta} - \mu (\mathbf{S}_{i,:} \mathbf{1}_{n_t} - 1) - \mathbf{S}_{i,:} \boldsymbol{\eta}^T \quad (17)$$

where μ and $\boldsymbol{\eta}$ are the Lagrangian multipliers.

To explore the data locality and reduce computation time, we prefer to learn a sparse $\mathbf{S}_{i,:}$, i.e., only the k -nearest neighbors of each sample are preserved to be locally connected. Based on the KKT condition, Eq.(17) has a closed-form solution:

$$S_{ij} = \max(z - \frac{A_{ij}}{2\delta}, 0), \quad z = \frac{1}{k} + \frac{1}{2k\delta} \sum_{j=1}^k \tilde{A}_{ij} \quad (18)$$

where \tilde{A}_{ij} is the element of matrix $\tilde{\mathbf{A}}$, obtained by sorting the entries for each row of \mathbf{A} in an ascending order. According to [21], we define $B_{ij} = \|\mathbf{x}_{ti} - \mathbf{x}_{tj}\|_2^2$ and set the value of parameter δ as:

$$\delta = \frac{1}{n_t} \sum_{i=1}^{n_t} \left(\frac{k}{2} \tilde{B}_{i,k+1} - \frac{1}{2} \sum_{j=1}^k \tilde{B}_{ij} \right) \quad (19)$$

Similar to \tilde{A}_{ij} , we also define \tilde{B}_{ij} as the element of matrix $\tilde{\mathbf{B}}$ which is obtained by sorting the entries for each row of \mathbf{B} from small to large.

We use a linear SVM¹ classifier to initialize the target label matrix. The initial adjacency matrix is obtained by solving each subproblem like (16) in the original space. The detailed optimization steps of CMMS are summarized in Algorithm 1.

¹<https://www.csie.ntu.edu.tw/~cjlin/liblinear/>

D. Convergence and Complexity Analysis

1) *Convergence Analysis*: We can prove the convergence of the proposed Algorithm 1 via the following proposition:

Proposition 1. *The proposed iterative optimization steps shown in algorithm 1 monotonically decreases the objective function value of (9) in each iteration.*

Proof. Assume that at the r -th iteration, we get $\mathbf{P}_r, \mathbf{F}_r, \mathbf{G}_r, \mathbf{S}_r$. We denote the value of the objective function in (9) at the r -th iteration as $f(\mathbf{P}_r, \mathbf{F}_r, \mathbf{G}_r, \mathbf{S}_r)$. In our Algorithm 1, we divide problem (9) into four subproblems (10), (12), (14) and (15), and each of them is a convex problem with respect to their corresponding variables. By solving the subproblems alternatively, our proposed algorithm can ensure finding the optimal solution of each subproblem, i.e., $\mathbf{P}_{r+1}, \mathbf{F}_{r+1}, \mathbf{G}_{r+1}, \mathbf{S}_{r+1}$. Therefore, as the combination of four subproblems, the objective function value of (9) in the $(r+1)$ -th iteration satisfies:

$$f(\mathbf{P}_r, \mathbf{F}_r, \mathbf{G}_r, \mathbf{S}_r) \geq f(\mathbf{P}_{r+1}, \mathbf{F}_{r+1}, \mathbf{G}_{r+1}, \mathbf{S}_{r+1}) \quad (20)$$

In light of this, the proof is completed and the algorithm will converge to local solution at least. \square

2) *Complexity Analysis*: The optimization Algorithm 1 of our CMMS comprises four subproblems. The complexity of these four subproblems are induced as follows: First, the cost of initializing \mathbf{S} is $\mathcal{O}(mn_t^2 + n_t^2 \log(n_t))$ and we ignore the time to initialize \mathbf{G}_t since the base classifier is very fast. Then, in each iteration, the complexity to construct and solve the generalized eigenvalue problem (12) for \mathbf{P} is $\mathcal{O}(n^2C + nC^2 + C^3 + mn^2 + nm^2 + dm^2)$. The target cluster centroids \mathbf{F} can be obtained with a time cost of $\mathcal{O}(dmn + dnC + dC^2)$. The complexity of updating the target labels matrix \mathbf{G}_t is $\mathcal{O}(Cdn_t)$. The adjacency matrix \mathbf{S} is updated with the cost of $\mathcal{O}(dn_t^2 + n_t^2 \log(n_t))$. Generally, we have $C < d < m$. Therefore, the overall computational complexity is $\mathcal{O}(n_t^2 \log(n_t)(T+1) + n^2mT + nm^2T + dm^2T)$, where T is the number of iteration.

IV. SEMI-SUPERVISED EXTENSION

In this section, we further extend our CMMS to semi-supervised domain adaptation including both homogeneous and heterogeneous settings.

1) *Homogeneous Setting*: We denote the target data as $\mathbf{X}_t = \{\mathbf{X}_l, \mathbf{X}_u\}$ where $\mathbf{X}_l = \{\mathbf{x}_{li}\}_{i=1}^{n_l}$ is the labeled data with the corresponding labels denoted by $\mathbf{Y}_l = \{\mathbf{y}_{li}\}_{i=1}^{n_l}$ and $\mathbf{X}_u = \{\mathbf{x}_{uj}\}_{j=1}^{n_u}$ is the unlabeled data. In the SDA scenario, except for the class centroids of source data, the few but precise labeled target data can provide additional valuable reference for determining the cluster centroids \mathbf{F} of unlabeled data. In this paper, we provide a simple but effective strategy to adaptively combine these two kinds of information. Specifi-

cally, our proposed semi-supervised extension is formulated as:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{F}, \mathbf{G}_u, \mathbf{S}} & \|\lambda_1 \mathbf{P}^T \mathbf{X}_s \mathbf{E}_s + \lambda_2 \mathbf{P}^T \mathbf{X}_l \mathbf{E}_l - \mathbf{F}\|_F^2 + \beta \|\mathbf{P}\|_F^2 \\ & + \alpha \|\mathbf{P}^T \mathbf{X}_t - \mathbf{F} \mathbf{G}_t^T\|_F^2 + \gamma (\text{tr}(\mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2) \\ \text{s.t. } & \lambda_1 + \lambda_2 = 1, \lambda_1, \lambda_2 \geq 0, \mathbf{P}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{P} = \mathbf{I}_d, \\ & \mathbf{G}_u \in \{0, 1\}^{n_t \times C}, \mathbf{S} \mathbf{1}_{n_t} = \mathbf{1}_{n_t}, 0 \leq S_{ij} \leq 1 \end{aligned} \quad (21)$$

where $n_t = n_l + n_u$, λ_1, λ_2 are balanced factors and $\mathbf{G}_t = [\mathbf{G}_l; \mathbf{G}_u]$. \mathbf{G}_l has the same definition with \mathbf{G}_s . Eq.(21) can be transformed to the standard formula as Eq.(9):

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{F}, \mathbf{G}, \mathbf{S}} & \|\mathbf{P}^T \mathbf{X} \mathbf{E} - \mathbf{F}\|_F^2 + \alpha \|\mathbf{P}^T \mathbf{X} \mathbf{V} - \mathbf{F} \mathbf{G}^T\|_F^2 \\ & + \beta \|\mathbf{P}\|_F^2 + \gamma (\text{tr}(\mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P}) + \delta \|\mathbf{S}\|_F^2) \\ \text{s.t. } & \lambda_1 + \lambda_2 = 1, \lambda_1, \lambda_2 \geq 0, \mathbf{P}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{P} = \mathbf{I}_d, \\ & \mathbf{G} \in \{0, 1\}^{n \times C}, \mathbf{S} \mathbf{1}_{n_t} = \mathbf{1}_{n_t}, 0 \leq S_{ij} \leq 1 \end{aligned} \quad (22)$$

where $\mathbf{E} = [\lambda_1 \mathbf{E}_s; \lambda_2 \mathbf{E}_l; \mathbf{0}_{n_u \times C}]$, $\mathbf{G} = [\mathbf{0}_{n_s \times C}; \mathbf{G}_t]$. In addition to the balanced factors λ_1 and λ_2 , the other variables in Eq.(22) can be readily solved with our Algorithm 1. Since the objective function is convex with respect to λ_1 and λ_2 , they can be solved easily with the closed-form solution: $\lambda_1 = \max(\min(\text{tr}(\mathbf{J} \mathbf{M}^T) / \text{tr}(\mathbf{J}^T \mathbf{J}), 1), 0)$, $\lambda_2 = 1 - \lambda_1$, where $\mathbf{J} = \mathbf{P}^T \mathbf{X}_s \mathbf{E}_s - \mathbf{P}^T \mathbf{X}_l \mathbf{E}_l$, $\mathbf{M} = \mathbf{F} - \mathbf{P}^T \mathbf{X}_t \mathbf{E}_l$.

2) *Heterogeneous Setting*: In the heterogeneous setting, the source and target data usually own different feature dimensions. Our proposed Eq.(21) can be naturally extended to the heterogeneous manner, only by replacing the projection matrix \mathbf{P} with two separate ones [26]:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{F}, \mathbf{G}_u, \mathbf{S}} & \|\lambda_1 \mathbf{P}_s^T \mathbf{X}_s \mathbf{E}_s + \lambda_2 \mathbf{P}_t^T \mathbf{X}_l \mathbf{E}_l - \mathbf{F}\|_F^2 \\ & + \gamma (2\text{tr}(\mathbf{P}_s^T \mathbf{X}_s \mathbf{L}_s \mathbf{X}_s^T \mathbf{P}_s) + 2\text{tr}(\mathbf{P}_t^T \mathbf{X}_t \mathbf{L}_t \mathbf{X}_t^T \mathbf{P}_t) + \\ & \delta \|\mathbf{S}\|_F^2) + \alpha \|\mathbf{P}_t^T \mathbf{X}_t - \mathbf{F} \mathbf{G}_t^T\|_F^2 + \beta (\|\mathbf{P}_s\|_F^2 + \|\mathbf{P}_t\|_F^2) \\ \text{s.t. } & \lambda_1 + \lambda_2 = 1, \lambda_1, \lambda_2 \geq 0, \mathbf{Z} \mathbf{H} \mathbf{Z}^T = \mathbf{I}_d, \\ & \mathbf{G}_t \in \{0, 1\}^{n_t \times C}, \mathbf{S} \mathbf{1}_{n_t} = \mathbf{1}_{n_t}, 0 \leq S_{ij} \leq 1 \end{aligned} \quad (23)$$

where $\mathbf{Z} = [\mathbf{P}_s^T \mathbf{X}_s, \mathbf{P}_t^T \mathbf{X}_t]$ is the new feature representations of two domains with the same dimension. α controls the effect of K -means clustering for target data and a proper value helps to achieve a good clustering performance. γ determines the influence of discriminative information preservation for source data and local manifold self-learning for target data, which is crucial to the equality improvement of class centroids for two domains. By defining $\mathbf{X} = \text{diag}(\mathbf{X}_s, \mathbf{X}_t)$, $\mathbf{P} = [\mathbf{P}_s; \mathbf{P}_t]$, Eq.(23) can be transformed to the standard formula as Eq.(22), and thus can be solved with the same algorithm.

V. EXPERIMENTS

In this section, we first describe all involved datasets. Next, the details of experimental setup including comparison methods in UDA and SDA scenarios, training protocol and parameter setting are given. Then, the experimental results in UDA scenario, ablation study, parameter sensitivity and convergence analysis are presented. Finally, we show the results in SDA scenario. The source code of this paper is available at <https://github.com/LeiTian-qj/CMMS/tree/master>.

A. Datasets and Descriptions

We apply our method to seven benchmark datasets which are widely used in domain adaptation. These datasets are represented with different kinds of features including Alexnet-FC₇, SURF, DeCAF₆, pixel, Resnet50 and BoW. Table II shows the overall descriptions of these datasets. We will introduce them in detail as follows.

Office31 [53] contains 4,110 images of office objects in 31 categories from three domains: Amazon (A), DSLR (D) and Webcam (W). Amazon images are downloaded from the online merchants. The images from DSLR domain are captured by a digital SLR camera while those from Webcam domain by a web camera. We adopt the AlexNet-FC₇ features² fine-tuned on source domain. Following [31], we have 6 cross-domain tasks, i.e., “A→D”, “A→W”, ..., “W→D”.

Office-Caltech10 [10] includes 2,533 images of objects in 10 shared classes between Office31 dataset and the Caltech256 (C) dataset. The Caltech256 dataset is a widely used benchmark for object recognition. We use the 800-dim SURF features³ and 4,096-dim DeCAF₆ features⁴ [54]. Following [9], we construct 12 cross-domain tasks, i.e., “A→C”, “A→D”, ..., “W→D”.

MSRC-VOC2007 [27] consists of two subsets: MSRC (M) and VOC2007 (V). It is constructed by selecting 1,269 images in MSRC and 1,530 images in VOC2007 which share 6 semantic categories: aeroplane, bicycle, bird, car, cow, sheep. We utilize the 256-dim pixel features⁵. Finally, we establish 2 tasks, “M→V” and “V→M”.

MNIST-USPS is made up of two handwritten digit image datasets: MNIST (Mn) and USPS (Us). Following [41], we sample 2,000 images from MNIST and 1,800 images from USPS. Besides, we utilize 2 conv-layer LeNet pre-trained on labeled source data to extract CNN features. Two tasks are obtained, i.e., “Mn→Us” and “Us→Mn”.

Office-Home [55] involves 15,585 images of daily objects in 65 shared classes from four domains: Art (artistic depictions of objects, Ar), Clipart (collection of clipart images, Cl), Product (images of objects without background, Pr) and Real-World (images captured with a regular camera, Re). We use the 4,096-dim Resnet50 features⁶ released by [33]. Similarly, we obtain 12 tasks, i.e., “Ar→Cl”, “Ar→Pr”, ..., “Re→Pr”.

Visda2017 [56] is a large-scale dataset for synthetic images to real images cross-domain task, which is first presented in 2017 Visual Domain Adaptation Challenge. Following [41], we take the the training data (152,397 synthetic images) as source domain, and the the validation data (55,388 real images) as target domain. The source and target domains share 12 object categories, i.e., aeroplane, bicycle, bus, car, horse, knife, motorcycle, person, plant, skateboard, train and trunk. We employ Resnet50 model pre-trained on ImageNet in Pytorch to extracted features.

Multilingual Reuters Collection [57] is a cross-lingual text dataset with about 11,000 articles from six common classes in

TABLE II
STATISTICS OF THE SEVEN BENCHMARK DATASETS

| Dataset | Subsets (Abbr.) | Samples | Feature (Size) | Classes |
|---------------------------------|-----------------|---------|--|---------|
| Office31 | Amazon (A) | 2,817 | Alexnet-FC ₆ (4,096) | 31 |
| | DSLR (D) | 498 | | |
| | Webcam (W) | 795 | | |
| Office-Caltech10 | Amazon (A) | 958 | SURF (800) DeCAF ₆ (4,096) | 10 |
| | Caltech (W) | 1,123 | | |
| | DSLR (D) | 157 | | |
| MSRC-VOC2007 | MSRC (M) | 1,269 | Pixel (256) | 6 |
| | VOC2007 (V) | 1,530 | | |
| MNIST-USPS | MNIST (Mn) | 2,000 | CNN features (2048) | 10 |
| | USPS (Us) | 1,800 | | |
| Office-Home | Art (Ar) | 2,427 | Resnet50 (2,048) | 65 |
| | Clipart (Cl) | 4,365 | | |
| | Product (Pr) | 4,439 | | |
| | RealWorld (Re) | 4,357 | | |
| Visda2017 | train | 152,397 | Resnet50 (2,048) | 12 |
| | validation | 55,388 | | |
| Multilingual Reuters Collection | English | 18,758 | BoW (1,131) | 6 |
| | French | 26,648 | BoW (1,230) | |
| | German | 29,953 | BoW (1,417) | |
| | Italian | 24,039 | BoW (1,041) | |
| | Spanish | 12,342 | BoW (807) | |

five languages: English, French, German, Italian, and Spanish. All articles are sampled by BoW features⁷ with TF-IDF. Then, they are processed by PCA for dimension reduction and the reduced dimensionality for English, French, German, Italian and Spanish are 1,131, 1,230, 1,417, 1,041 and 807, respectively. We pick the Spanish as the target and each of the rest as the source by turns. Eventually, we gain four tasks.

B. Experimental Setup

1) *Comparison Methods in UDA Scenario*: 1-NN, SVM¹, GFK [10], JDA [9], CORAL [12], DCD [16], JGSA [58], DICE [31], MEDA [29], SPL [33], MCS [32], EasyTL [59], CAPLS [60], TADA [61], ADDA [37], the method of [38], TPN [41], CADA-P [62], DCAN [63], DREMA [64], GVB-GD [65], RSDA-DANN [66], DSAN [67], SimNet [39], the method of [68], 3CATN [69], DTA [70], DM-ADA [71], ALDA [72].

2) *Comparison Methods in SDA Scenario*: SVM_t, SVM_{st}¹, MMDT [22], DTMKL [73], CDLS [43], ILS [23], TFMKL-S and TFMKL-H [24], SHFA [25], The method of Li *et al.* [42]. SVM_t, SVM_{st}, MMDT, DTMKL-f, TFMKL-S and TFMKL-H are employed in the homogeneous setting while SVM_t, MMDT, SHFA, CDLS, the method of [42] in the heterogeneous setting and CDSPP [74].

3) *Training Protocol*: For UDA scenario, all source samples are utilized for training like [16]. We exploit *z*-score standardization [10] on all kinds of features. For SDA scenario, in homogeneous setting, we use the Office-Caltech10 and MSRC-VOC2007 datasets following the same protocol with [24]. Specifically, for the Office-Caltech10 dataset, we randomly choose 20 samples per category for amazon domain while 8 for the others as the sources. Three labeled target samples per class are selected for training while the rest for testing. For fairness, we use the train/test splits released by [22]. For the MSRC-VOC2007 dataset, all source samples are utilized for training, and 2 or 4 labeled target samples per category are randomly selected for training with the remaining to be recognized. In heterogeneous setting, we employ the

²<https://github.com/VisionLearningGroup/CORAL/tree/master/dataset>

³<http://boqinggong.info/assets/GFK.zip>

⁴<https://github.com/jindongwang/transferlearning/blob/master/data/>

⁵<http://ise.thss.tsinghua.edu.cn/~mlong/>

⁶<https://github.com/hellowangqian/domainadaptation-caps>

⁷<http://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual,+Multiview+Text+Categorization+Test+collection>

TABLE III
CLASSIFICATION ACCURACIES (%) ON OFFICE31 DATASET ($\alpha = 0.1, \beta = 0.1$)

| Task | 1-NN | SVM | GFK | JDA | CORAL | DICD | JGSA | DICE | MEDA | EasyTL | CAPLS | SPL | MCS | CMMS |
|---------|------|------|------|------|-------|------|-------------|--------------|------|--------|--------------|-------------|-------------|-------------|
| A→D | 59.8 | 58.8 | 61.8 | 65.5 | 65.7 | 66.5 | 69.5 | 67.5 | 69.5 | 65.3 | 69.9 | 69.1 | 71.9 | 72.9 |
| A→W | 56.4 | 58.9 | 58.9 | 70.6 | 64.3 | 73.3 | 70.4 | 71.9 | 69.9 | 66.0 | 68.2 | 69.9 | 75.1 | 74.7 |
| D→A | 38.1 | 48.8 | 45.7 | 53.7 | 48.5 | 56.3 | 56.6 | 57.8 | 58.0 | 50.5 | 56.5 | 62.5 | 58.8 | 60.7 |
| D→W | 94.7 | 95.7 | 96.4 | 98.2 | 96.1 | 96.9 | 98.2 | 97.2 | 94.0 | 93.3 | 98.0 | 97.7 | 96.7 | 97.6 |
| W→A | 39.8 | 47.0 | 45.5 | 52.1 | 48.2 | 55.9 | 54.2 | 60.0 | 56.0 | 50.6 | 58.6 | 58.2 | 57.2 | 60.3 |
| W→D | 98.4 | 98.2 | 99.6 | 99.2 | 99.8 | 99.4 | 99.2 | 100.0 | 96.8 | 97.2 | 100.0 | 99.6 | 99.4 | 99.6 |
| Average | 64.5 | 67.9 | 68.0 | 73.4 | 70.4 | 74.7 | 74.7 | 75.7 | 74.0 | 70.5 | 75.2 | 76.2 | 76.5 | 77.6 |

TABLE IV
CLASSIFICATION ACCURACIES (%) ON OFFICE-CALTECH10 DATASET WITH SURF FEATURES ($\alpha = 0.1, \beta = 0.2$)

| Task | 1-NN | SVM | GFK | JDA | CORAL | DICD | JGSA | DICE | MEDA | EasyTL | CAPLS | SPL | MCS | CMMS |
|---------|------|------|------|------|-------------|------|-------------|-------------|-------------|-------------|-------|-------------|-------------|-------------|
| A→C | 26.0 | 35.6 | 41.0 | 39.4 | 45.1 | 42.4 | 41.5 | 42.7 | 43.9 | 42.3 | 43.0 | 41.2 | 44.1 | 39.4 |
| A→D | 25.5 | 36.3 | 40.7 | 39.5 | 39.5 | 38.9 | 47.1 | 49.7 | 45.9 | 48.4 | 40.8 | 44.6 | 55.4 | 53.5 |
| A→W | 29.8 | 31.9 | 41.4 | 38.0 | 44.4 | 45.1 | 45.8 | 52.2 | 53.2 | 43.1 | 42.0 | 58.0 | 40.3 | 56.3 |
| C→A | 23.7 | 42.9 | 40.2 | 44.8 | 54.3 | 47.3 | 51.5 | 50.2 | 56.5 | 52.6 | 53.1 | 53.3 | 53.9 | 61.0 |
| C→D | 25.5 | 33.8 | 40.0 | 45.2 | 36.3 | 49.7 | 45.9 | 51.0 | 50.3 | 50.6 | 41.4 | 41.4 | 46.5 | 51.0 |
| C→W | 25.8 | 34.6 | 36.3 | 41.7 | 38.6 | 46.4 | 45.4 | 48.1 | 53.9 | 53.9 | 43.7 | 61.7 | 54.2 | 61.7 |
| D→A | 28.5 | 34.3 | 30.7 | 33.1 | 37.7 | 34.5 | 38.0 | 41.1 | 41.2 | 38.3 | 34.0 | 35.3 | 38.3 | 46.7 |
| D→C | 26.3 | 32.1 | 31.8 | 31.5 | 33.8 | 34.6 | 29.9 | 33.7 | 34.9 | 36.1 | 27.1 | 25.9 | 31.6 | 31.9 |
| D→W | 63.4 | 78.0 | 87.9 | 89.5 | 84.7 | 91.2 | 91.9 | 84.1 | 87.5 | 86.1 | 82.7 | 82.7 | 85.4 | 86.1 |
| W→A | 23.0 | 37.5 | 30.1 | 32.8 | 35.9 | 34.1 | 39.9 | 37.5 | 42.7 | 38.2 | 36.3 | 41.1 | 37.3 | 40.1 |
| W→C | 19.9 | 33.9 | 32.0 | 31.2 | 33.7 | 33.6 | 33.2 | 37.8 | 34.0 | 35.4 | 33.1 | 37.8 | 33.8 | 35.8 |
| W→D | 59.2 | 80.9 | 84.4 | 89.2 | 86.6 | 89.8 | 90.5 | 87.3 | 88.5 | 79.6 | 80.3 | 83.4 | 80.9 | 89.2 |
| Average | 31.4 | 42.6 | 44.7 | 46.3 | 47.6 | 49.0 | 50.0 | 51.3 | 52.7 | 50.5 | 46.5 | 50.5 | 50.1 | 54.4 |

Office-Caltech10 and Multilingual Reuters Collection datasets using the experiment setting of [42]. For the Office-Caltech10 dataset, the SURF and DeCAF₆ features are served as the source and target. The source domain contains 20 instances per class, and 3 labeled target instances per category are selected for training with the rest for testing. For the Multilingual Reuters Collection dataset, Spanish is selected as the target and the remaining as the source by turns. 100 articles per category are randomly selected to build the source domain, and 10 labeled target articles per category are selected for training with 500 articles per class from the rest to be classified.

4) *Parameter Setting*: In both UDA and SDA scenarios, we do not have massive labeled target samples, so we can not perform a standard cross-validation procedure to obtain the optimal parameters. For a fair comparison, we cite the results from the original papers or run the code provided by the authors. Following [16], we grid-search the hyperparameter space and report the best results. For GFK, JDA, DICD, JGSA, DICE and MEDA, the optimal reduced dimension is searched in $d \in \{10, 20, \dots, 100\}$. The best value of regularization parameter for projection is searched in the range of $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0\}$. For the two recent methods, SPL and MCS, we adopt the default parameters used in their public codes or follow the procedures for tuning parameters according to the corresponding original papers. For our method, we fix $d = 100$, $\gamma = 5.0$ and $k = 10$ leaving α , β tunable. We obtain the optimal parameters by searching $\alpha, \beta \in [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0]$.

C. Unsupervised Domain Adaptation

1) *Experimental results on UDA*: **Results on Office31 dataset**. Table III summarizes the classification results on the Office31 dataset, where the highest accuracy of each cross-domain task is boldfaced. We can observe that CMMS has the best average performance, with a 1.1% improvement over

the optimal competitor MCS. CMMS achieves the highest results on 2 out of 6 tasks, while MCS only works the best for task A→W with just 0.4% higher than CMMS. Generally, SPL, MCS and CMMS perform better than those methods that classify target samples independently, which demonstrates that exploring the structure information of data distribution can facilitate classification performance. However, compared with SPL and MCS, CMMS further mines and exploits the inherent local manifold structure of target data to promote cluster prototypes learning, thus can lead to a better performance.

Results on Office-Caltech10 dataset. The results on Office-Caltech10 dataset with SURF features are listed in Table IV. Regarding the average accuracy, CMMS shows a large advantage which improves 1.7% over the second best method MEDA. CMMS is the best method on 4 out of 12 tasks, while MEDA only wins two tasks. On C→A, D→A and C→W, CMMS leads MEDA by over 4.5% margin. Following [33], we also employ the DeCAF₆ features, and the classification results are shown in Table V. CMMS is superior to all competitors with regard to the average accuracy and works the best or second best for all tasks except for C→W. Carefully comparing the results of SURF features and DeCAF₆ features, we can find that SPL and MCS prefer to deep features. Nevertheless, CMMS does not have such a preference, which illustrates that CMMS owns better generalization capacity.

Results on MSRC-VOC2007 and MNIST-USPS datasets. The experimental results on the MSRC-VOC2007 and MNIST-USPS datasets are reported in Table VI. On MSRC-VOC2007 dataset, the average classification accuracy of CMMS is 55.4%, which is significant higher than those of all competitors. Especially, on task V→M, CMMS gains a huge performance improvement of 15.3% compared with the second best method SPL, which verifies the significant effectiveness of our proposal. On MNIST-USPS dataset, CMMS achieves the highest average classification accuracy among all methods,

TABLE V
CLASSIFICATION ACCURACIES (%) ON OFFICE-CALTECH10 DATASET WITH DECAF₆ FEATURES ($\alpha = 0.2, \beta = 0.5$)

| Task | 1-NN | SVM | GFK | JDA | CORAL | DICD | JGSA | DICE | MEDA | EasyTL | CAPLS | SPL | MCS | CMMS |
|---------|------|------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|--------|--------------|--------------|-------------|--------------|
| A→C | 71.7 | 84.4 | 77.3 | 83.2 | 83.2 | 86.0 | 84.9 | 85.9 | 87.4 | 86.5 | 86.1 | 87.4 | 88.3 | 88.8 |
| A→D | 73.9 | 83.4 | 84.7 | 86.6 | 84.1 | 83.4 | 88.5 | 89.8 | 88.1 | 91.7 | 94.9 | 89.2 | 91.7 | 95.5 |
| A→W | 68.1 | 76.9 | 81.0 | 80.3 | 74.6 | 81.4 | 81.0 | 86.4 | 88.1 | 85.8 | 87.1 | 95.3 | 91.5 | 92.2 |
| C→A | 87.3 | 91.3 | 88.5 | 88.7 | 92.0 | 91.0 | 91.4 | 92.3 | 93.4 | 93.0 | 90.8 | 92.7 | 93.5 | 94.1 |
| C→D | 79.6 | 85.4 | 86.0 | 91.1 | 84.7 | 93.6 | 93.6 | 93.6 | 91.1 | 89.2 | 95.5 | 98.7 | 90.4 | 95.5 |
| C→W | 72.5 | 77.3 | 80.3 | 87.8 | 80.0 | 92.2 | 86.8 | 93.6 | 95.6 | 82.7 | 85.4 | 93.2 | 85.1 | 91.9 |
| D→A | 49.9 | 86.5 | 85.8 | 91.8 | 85.5 | 92.2 | 92.0 | 92.5 | 93.2 | 91.3 | 93.0 | 92.9 | 93.5 | 93.4 |
| D→C | 42.0 | 77.1 | 76.0 | 85.5 | 76.8 | 86.1 | 86.2 | 87.4 | 87.5 | 84.5 | 88.8 | 88.6 | 89.0 | 89.3 |
| D→W | 91.5 | 99.3 | 97.3 | 99.3 | 99.3 | 99.0 | 99.7 | 90.0 | 97.6 | 98.0 | 100.0 | 98.6 | 99.3 | 99.3 |
| W→A | 62.5 | 80.7 | 81.8 | 90.2 | 81.2 | 89.7 | 90.7 | 90.7 | 99.4 | 89.8 | 92.3 | 92.0 | 93.4 | 93.8 |
| W→C | 55.3 | 72.5 | 73.9 | 84.2 | 75.5 | 84.0 | 85.0 | 85.3 | 93.2 | 80.8 | 88.2 | 87.0 | 88.3 | 89.0 |
| W→D | 98.1 | 99.4 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 99.4 | 99.4 | 100.0 | 100.0 | 100.0 | 100.0 |
| Average | 71.0 | 84.5 | 83.2 | 89.1 | 84.7 | 89.9 | 90.0 | 91.4 | 92.8 | 89.4 | 91.8 | 93.0 | 92.0 | 93.6 |

TABLE VI
CLASSIFICATION ACCURACIES (%) ON MSRC-VOC2007 DATASET (TOP, $\alpha = 0.1, \beta = 0.05$) AND MNIST-USPS DATASET (BELOW, $\alpha = 0.2, \beta = 0.05$)

| Task | 1-NN | SVM | GFK | JDA | CORAL | DICD | JGSA | DICE | MEDA | EasyTL | CAPLS | SPL | MCS | CMMS |
|---------|------|------|------|------|-------------|------|------|------|------|--------|-------|------|------|-------------|
| M→V | 35.5 | 35.6 | 34.7 | 30.4 | 38.4 | 32.4 | 35.2 | 33.1 | 35.3 | 33.1 | 32.5 | 34.7 | 31.8 | 31.8 |
| V→M | 47.2 | 51.8 | 48.9 | 44.8 | 54.9 | 47.8 | 47.5 | 46.3 | 60.1 | 58.3 | 49.2 | 63.8 | 66.5 | 79.1 |
| Average | 41.3 | 43.7 | 41.8 | 37.6 | 46.7 | 40.1 | 41.3 | 39.7 | 47.7 | 45.7 | 40.9 | 49.3 | 49.2 | 55.4 |

| Task | 1-NN | SVM | JDA | DICD | JGSA | DICE | MEDA | EasyTL | SPL | MCS | ADDA | TPN | Chadha [38] | CMMS |
|---------|------|------|------|------|------|------|------|--------|-------------|------|------|------|-------------|-------------|
| Mn→Us | 90.5 | 88.6 | 93.2 | 94.4 | 93.4 | 94.9 | 94.7 | 90.3 | 95.4 | 94.9 | 89.4 | 92.1 | 92.5 | 95.4 |
| Us→Mn | 88.8 | 86.6 | 92.0 | 92.2 | 91.3 | 92.0 | 93.6 | 90.5 | 94.1 | 93.8 | 90.1 | 94.1 | 96.7 | 94.3 |
| Average | 89.7 | 87.6 | 92.6 | 93.3 | 92.3 | 93.4 | 94.2 | 90.4 | 94.7 | 94.3 | 89.8 | 93.1 | 94.6 | 94.8 |

TABLE VII
CLASSIFICATION ACCURACIES (%) ON OFFICE-HOME DATASET ($\alpha = 0.1, \beta = 0.01$). DEEP LEARNING METHODS ARE LISTED BELOW CMMS. THE HIGHEST ACCURACY OF EACH CROSS-DOMAIN TASK WITH TRADITIONAL/DEEP LEARNING METHODS IS BOLD FACED.

| Methods | Ar→Cl | Ar→Pr | Ar→Re | Cl→Ar | Cl→Pr | Cl→Re | Pr→Ar | Pr→Cl | Pr→Re | Re→Ar | Re→Cl | Re→Pr | Average |
|---------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1-NN | 37.9 | 54.4 | 61.6 | 40.7 | 52.7 | 52.5 | 47.1 | 41.1 | 66.7 | 57.1 | 45.1 | 72.9 | 52.5 |
| SVM | 42.4 | 61.2 | 69.9 | 42.6 | 56.2 | 57.7 | 48.7 | 41.5 | 70.6 | 61.6 | 45.7 | 76.1 | 56.2 |
| 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 | 53.6 |
| 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 |
| CORAL | 47.3 | 69.3 | 74.6 | 54.2 | 67.2 | 67.8 | 55.7 | 43.0 | 73.9 | 64.2 | 49.2 | 78.0 | 62.0 |
| 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 |
| JGSA | 51.3 | 72.9 | 78.5 | 58.1 | 72.4 | 73.4 | 62.3 | 50.3 | 79.4 | 67.9 | 53.4 | 80.4 | 63.3 |
| DICE | 49.1 | 70.7 | 73.9 | 51.4 | 65.9 | 65.9 | 60.0 | 48.6 | 76.2 | 65.4 | 53.5 | 78.8 | 63.3 |
| MEDA | 52.1 | 75.3 | 77.6 | 61.0 | 76.5 | 76.8 | 61.8 | 53.4 | 79.5 | 68.1 | 55.1 | 82.5 | 68.3 |
| EasyTL | 49.8 | 72.5 | 75.8 | 60.7 | 69.5 | 71.2 | 59.0 | 47.1 | 76.4 | 64.8 | 51.1 | 77.3 | 64.6 |
| CAPLS | 56.2 | 78.3 | 80.2 | 66.0 | 75.4 | 78.4 | 66.4 | 53.2 | 81.1 | 71.6 | 56.1 | 84.3 | 70.6 |
| SPL | 54.5 | 77.8 | 81.9 | 65.1 | 78.0 | 81.1 | 66.0 | 53.1 | 82.8 | 69.9 | 55.3 | 86.0 | 71.0 |
| MCS | 53.8 | 78.4 | 78.8 | 64.0 | 75.0 | 78.9 | 64.8 | 52.3 | 79.9 | 67.0 | 55.8 | 80.3 | 69.1 |
| CMMS | 56.2 | 80.8 | 82.8 | 65.9 | 78.7 | 82.2 | 67.7 | 54.5 | 82.9 | 69.5 | 57.1 | 85.2 | 72.0 |

| | | | | | | | | | | | | | |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| TADA | 53.1 | 72.3 | 77.2 | 59.1 | 71.2 | 72.1 | 59.7 | 53.1 | 78.4 | 72.4 | 60.0 | 82.9 | 67.6 |
| CADA-P | 56.9 | 76.4 | 80.7 | 61.3 | 75.2 | 75.2 | 63.2 | 54.5 | 80.7 | 73.9 | 61.5 | 84.1 | 70.2 |
| DCAN | 54.5 | 75.7 | 81.2 | 67.4 | 74.0 | 76.3 | 67.4 | 52.7 | 80.6 | 74.1 | 59.1 | 83.5 | 70.5 |
| DRMEA | 52.3 | 73.0 | 77.3 | 64.3 | 72.0 | 71.8 | 63.6 | 52.7 | 78.5 | 72.0 | 57.7 | 51.6 | 68.1 |
| GVB-GD | 57.0 | 74.7 | 79.8 | 64.6 | 74.1 | 74.6 | 65.2 | 55.1 | 81.0 | 74.6 | 59.7 | 84.3 | 70.4 |
| RSDA-DANN | 51.5 | 76.8 | 81.1 | 67.1 | 72.1 | 77.0 | 64.2 | 51.1 | 81.8 | 74.9 | 55.9 | 84.5 | 69.8 |
| DSAN | 54.4 | 70.8 | 75.4 | 60.4 | 67.8 | 68.0 | 62.6 | 55.9 | 78.5 | 73.8 | 60.6 | 83.1 | 67.6 |

which confirms the superior generalization capacity of our proposal.

Results on Office-Home dataset. For fairness, we employ the deep features recently released by [33], which are extracted using the Resnet50 model pre-trained on ImageNet. Table VII summarizes the classification accuracies. For traditional methods, CMMS outperforms the second best method SPL in average performance, and achieves the best performance on 9 out of all 12 tasks while SPL only works the best for task Re→Pr with just 0.8% superiority to CMMS. This phenomenon shows that even the target samples are well clustered within the deep feature space, exploiting the inherent local manifold structure is still crucial to the improvement of the classification performance. Besides, we show the results of eight recent deep learning methods for comparison, which all adopt the Resnet50 model as the basic network structure. In terms of the average accuracy, CMMS performs better than all

deep methods with 1.5% improvement at least, which validates the significant superiority of our proposal.

Results on Visda2017 dataset. Visda2017 is a large-scale dataset that is published recently and has been widely employed by various deep learning methods. Following [41], we take the per category classification accuracy of target data and the mean of accuracy over all 12 categories as evaluation metric. All experiments are conducted with MATLAB R2016b on a standard Linux server with Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz CPU, 251GB RAM. The experimental results of six traditional methods and eleven deep methods are provided in Table VIII. The results of some other traditional methods are not reported due to the limited memory. We can observe that CMMS achieves the highest mean accuracy among all traditional and deep methods, which demonstrates that our CMMS has outstanding capability to deal with the domain adaptation tasks with more data samples.

TABLE VIII

CLASSIFICATION ACCURACIES (%) ON VISDA2017 DATASET ($\alpha = 0.1, \beta = 0.01$). DEEP LEARNING METHODS ARE LISTED BELOW OUR CMMS. * INDICATES THE RESULTS ARE OBTAINED USING RESNET-101 FEATURES OR RESNET-101 MODEL, AND — REPRESENTS THE RESULTS ARE UNAVAILABLE. THE RESULTS OF JGSA AND LPJT ARE CITED FROM [51]. THE HIGHEST ACCURACY OF EACH TASK WITH TRADITIONAL/DEEP LEARNING METHODS IS BOLDFACED.

| Method | aeroplane | bicycle | bus | car | horse | knife | motorcycle | person | plant | skateboard | train | truck | Mean |
|-----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1-NN | 90.0 | 55.2 | 66.1 | 43.0 | 56.8 | 26.2 | 91.4 | 31.4 | 35.3 | 31.4 | 70.6 | 36.6 | 52.8 |
| SVM | 77.1 | 41.8 | 73.5 | 53.3 | 53.2 | 21.5 | 82.9 | 34.9 | 57.2 | 51.6 | 83.5 | 24.9 | 50.2 |
| GFK | 88.5 | 55.9 | 67.9 | 42.4 | 65.3 | 30.0 | 90.6 | 36.4 | 42.0 | 32.3 | 73.3 | 34.1 | 54.9 |
| JDA | 91.2 | 73.2 | 67.4 | 52.5 | 88.9 | 67.6 | 85.1 | 69.0 | 80.7 | 48.1 | 81.6 | 31.9 | 69.8 |
| JGSA* | 91.1 | 63.8 | 64.2 | 47.5 | 92.6 | 42.3 | 79.8 | 76.5 | 84.3 | 50.1 | 72.2 | 26.5 | 65.8 |
| LPJT* | 93.0 | 80.3 | 66.5 | 56.3 | 95.8 | 70.3 | 74.2 | 83.8 | 91.7 | 40.0 | 78.7 | 57.6 | 74.0 |
| MCS | 94.9 | 80.6 | 79.1 | 56.7 | 92.8 | 94.8 | 82.9 | 70.0 | 84.4 | 91.3 | 81.4 | 43.1 | 79.3 |
| CMMS | 93.9 | 76.9 | 86.1 | 58.7 | 93.2 | 96.3 | 84.1 | 80.1 | 83.7 | 87.2 | 80.0 | 62.7 | 81.9 |
| SimNet | 94.5 | 80.2 | 69.5 | 43.5 | 89.5 | 16.6 | 76.0 | 81.1 | 86.4 | 76.4 | 79.6 | 41.9 | 69.6 |
| Wu <i>et al.</i> [68] | 91.5 | 80.4 | 82.3 | 59.2 | 90.1 | 62.9 | 85.3 | 75.0 | 87.3 | 70.6 | 78.5 | 28.4 | 74.3 |
| 3CATN | — | — | — | — | — | — | — | — | — | — | — | — | 73.2 |
| TPN | 93.7 | 85.1 | 69.2 | 81.6 | 93.5 | 61.9 | 89.3 | 81.4 | 93.5 | 81.6 | 84.5 | 49.9 | 80.4 |
| DTA | 93.1 | 70.5 | 83.8 | 87.0 | 92.3 | 3.3 | 91.9 | 86.4 | 93.1 | 71.0 | 82.0 | 15.3 | 76.2 |
| DM-ADA* | — | — | — | — | — | — | — | — | — | — | — | — | 75.6 |
| ALDA | 87.0 | 61.3 | 78.7 | 67.9 | 83.7 | 89.4 | 89.5 | 71.0 | 95.4 | 71.9 | 89.6 | 33.1 | 76.5 |
| GVB-GD | — | — | — | — | — | — | — | — | — | — | — | — | 75.3 |
| RSDA-DANN | — | — | — | — | — | — | — | — | — | — | — | — | 75.8 |
| DSAN* | 90.9 | 66.9 | 75.7 | 62.4 | 88.9 | 77.0 | 93.7 | 75.1 | 92.8 | 67.6 | 89.1 | 39.4 | 75.1 |

TABLE IX

THE CLASSIFICATION PERFORMANCE(%) OF CMMS, THE FIVE VARIANTS OF CMMS AND JDA.

| Dataset | JDA | CMMS _{cm} | CMMS _{rm} | CMMS _{pa} | CMMS _{op} | CMMS _{ds} | CMMS |
|---------------------------|------|--------------------|--------------------|--------------------|--------------------|--------------------|-------------|
| Office31 | 73.4 | 74.6 | 75.8 | 76.2 | 77.3 | 74.8 | 77.6 |
| Office-Caltech10(SURF) | 46.3 | 51.4 | 52.4 | 53.2 | 53.4 | 51.8 | 54.4 |
| Office-Caltech10 (DeCAF6) | 89.1 | 92.8 | 92.9 | 93.2 | 93.5 | 92.3 | 93.6 |
| MSRC-VOC2007 | 37.6 | 53.2 | 53.5 | 55.1 | 55.1 | 54.9 | 55.4 |
| MNIST-USPS | 92.6 | 93.9 | 93.8 | 94.4 | 94.4 | 93.6 | 94.8 |
| Office-Home | 61.9 | 70.2 | 70.7 | 71.2 | 71.7 | 70.6 | 72.0 |
| Visda2017 | 69.8 | 73.8 | 80.0 | 78.5 | 78.4 | 74.9 | 81.9 |
| Average | 67.2 | 72.8 | 74.2 | 74.5 | 74.8 | 73.3 | 75.7 |

2) *Ablation Study* : To understand our CMMS more deeply, we propose five variants of CMMS: **a)** CMMS_{cm}, only considers class Centroid Matching for two domains, i.e., the combination of Eq.(2), Eq.(3) and Eq.(7); **b)** CMMS_{rm}, does not utilize the local manifold structure of target data, i.e., removing Eq.(4) from our objective function Eq.(8); **c)** CMMS_{pa}, considers local manifold structure of target data by Predefining Adjacency matrix in the original feature space, i.e., replacing Eq.(4) with the Laplacian regularization; **d)** CMMS_{ds}, exploits the Discriminative Structure information of target domain via assigning pseudo-labels to target data and then minimizing the intra-class scatter in the projected space like Eq.(5); **e)** CMMS_{op}, carries out manifold learning withOut Projection, i.e., initializing target adjacency matrix in the original feature space using local manifold self-learning strategy and keeping it unchanged in the subsequent iterations. Table IX shows the results of CMMS and all variants. The results of classical JDA are also provided. Based on this table, we will analyze our approach in more detail as follows.

Effectiveness of class centroid matching. CMMS_{cm} consistently precedes JDA on six datasets, which confirms the remarkable superiority of our proposal to the MMD based pioneering approach. By the class centroid matching strategy, we can make full use of the structure information of data distribution, thus target samples are supposed to present favourable cluster distribution. To have a clear illustration, in Fig. 3, we display the *t*-SNE [75] visualization of the target features in the projected space on task V→M of MSRC-VOC2007 dataset. We can observe that JDA features are mixed

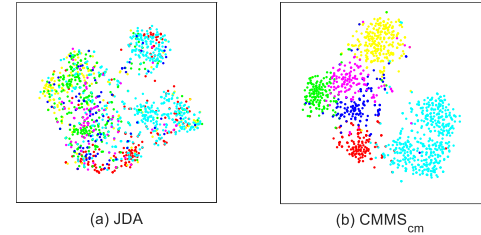


Figure 3. Target feature visualization of JDA and CMMS_{cm} on task V→M. Samples in different classes are presented by different colors.

together while CMMS_{cm} features are well-separated with cluster structure, which verifies the significant effectiveness of our class centroid matching strategy.

Effectiveness of local manifold self-learning strategy for target data. CMMS_{pa} performs better than CMMS_{rm} on nearly all datasets, which indicates that exploiting the local manifold structure of target samples help to classify them more successfully, even though the manifold structure is not so reliable. However, if we can capture it more faithfully, we can achieve a superior performance, which is verified by comparing CMMS with CMMS_{pa}. For a better understanding, we show the visualization of target adjacency matrix on task A→D (SURF) in Fig. 4. These matrices are obtained by either the self-learned distance or the predefined distances which include Euclidean distance, heatkernel distance with kernel width 1.0 and cosine distance. As we can see from Fig. 4, all predefined distances tend to incorrectly connect unrelated samples, and hardly capture the inherent local mani-

fold structure of target data. However, the self-learned distance can adaptively build the connections between intrinsic similar samples, thus can improve the classification performance. Moreover, CMMS_{op} is inferior to CMMS on all datasets, which indicates the necessity of operating manifold learning in the projected space. In such way, the curse of dimensionality could be relieved, and the inherent local manifold structure of target data can be captured more effectively. Generally, CMMS_{ds} performs much worse than CMMS and even worse than CMMS_{rm} which verifies that utilizing the discriminative information of target domain via assigning pseudo-labels to target samples independently is far from enough to achieve satisfactory results. The reason is that the pseudo-labels may be inaccurate and could cause error accumulation during learning, and thus the performance is degraded dramatically. In summary, our local manifold self-learning strategy can effectively enhance the utilization of structure information contained in target data.

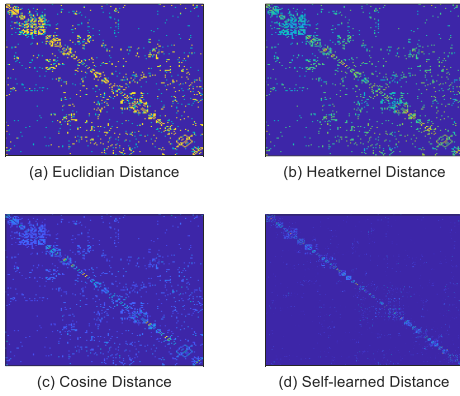


Figure 4. Target adjacency matrix visualization on task A→D (SURF) obtained by predefined or self-learned distances.

3) *Analytical Experiments:* In this part, we further conduct several experiments to pursue deeper insight for our CMMS approach.

Is the target prototype cluster assignment really equal to the target class? To validate whether the target prototype cluster assignment is really equal to the actual target class, in Fig. 5 (a), we plot the evolution of classification error for each class given target ground-truth labels over training iterations on task V→M. As we can see, for most target prototype clusters (5 out of 6), the trend of classification error meets the ideal situation, *i.e.*, it would decrease over iterations. However, we find that the prototype cluster 3 is an exception. To figure out the reason, in Table X, we summarize the sample size and proportion of each class for target data. For a more intuitive understanding, we further display the visualization result of target features in the projected space according to the prototype cluster pseudo-labels (Fig. 5 (d)) and the ground-truth labels (Fig. 5 (e)). By the comprehensive analysis of Fig. 5 (d), (e) and Table X, we argue that the potential reason could be that the sample size of target class 3 is very small and its sample distribution is easily mixed up with target class 6. In the training iteration process, more and more samples from target class 6 could be divided into target prototype cluster 3, leading to the centroid of target prototype cluster 3 deviates

TABLE X
SIZE AND PROPORTION (%) FOR EACH CLASS OF TARGET DATA (V→M).

| Class | 1 | 2 | 3 | 4 | 5 | 6 |
|-----------------|--------|----------|--------|----------|----------|----------|
| Size/Proportion | 58/4.6 | 272/21.4 | 72/5.7 | 495/39.0 | 182/14.3 | 190/15.0 |

from the actual centroid of target class 3. One promising solution may be that during each training iteration, we can first include the target samples with high-confidence pseudo-label for each category to train by some selective strategies, *e.g.*, self-paced learning, and then maximize the distances of different prototype clusters.

Is there a semantic mismatch between the target prototype cluster and its nearest source class? For a target prototype cluster, our CMMS assigns a pseudo-label l_i^{cmms} ($i = 1, \dots, C$) to it by searching the nearest class centroid of source data. When the ground-truth labels are given, the true-label of a target prototype cluster could be determined by the majority voting of its member's label and here we call it the "oracle label" (denoted as l_i^{oracle}). In Fig. 5 (b), we plot two sets of curves. The solid lines display the distance between each target prototype cluster centroid and its corresponding real source class centroid (*i.e.*, the source class with l_i^{oracle} label), while the dashed lines display the distance between each target prototype cluster centroid and its nearest source class centroid (*i.e.*, the source class with l_i^{cmms} label). Ideally, the solid line shall coincide with the dashed line for each target prototype cluster, which means an exact semantic match between the target prototype cluster and its nearest source class. From Fig. 5 (b), we observe that only target prototype cluster 3 semantically mismatches with source class 3. For the other 5 target prototype clusters, the solid lines overlap the dashed lines, and the downward trend of these lines indicates that our CMMS could improve their corresponding semantic match over the training iterations.

Does each target prototype cluster's members become homogeneous over training progress? In Fig. 5 (c), we display the evolution of homogeneity for each target prototype cluster on task V→M. It is obvious that all curves rise, which indicates that each target prototype cluster become homogeneous over training process. The homogeneity of 4 target clusters are more than 50%, which means that our proposal can produce high-purity prototype clusters. The low homogeneity of target prototype cluster 3 further confirm our aforementioned analysis for Fig. 5 (a), *i.e.*, the sample size of real target class 3 is very small and the samples of other target classes could be incorrectly divided into it.

How does our CMMS perform on imbalanced dataset? To evaluate the generalization ability of our CMMS on imbalanced dataset, an additional experiment is conducted on a new dataset. The source domain of this new dataset consists of Caltech10, dslr and webcam of Office-Caltech10 dataset. For its target domain, we design a new amazon domain. Specifically, for each category of the original amazon domain, we extract their first 82 images. The purpose of such operation is to force the sample proportion of each class to be equal in the new amazon domain. We choose the new amazon

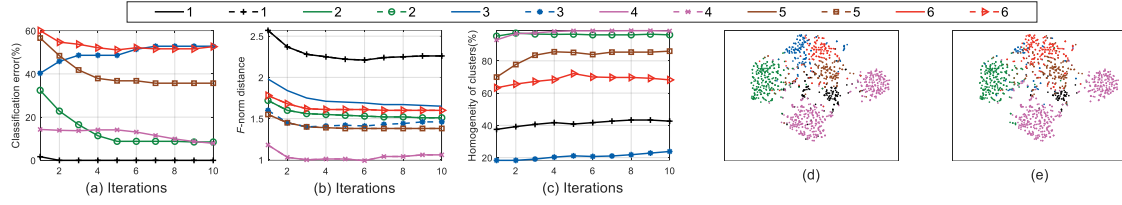


Figure 5. (a) Classification error given target ground-truth versus prototype cluster over training iterations on task $V \rightarrow M$. The number i indicates the i th prototype cluster. (b) The F -norm distance between each prototype cluster centroid and its source data correspondence class centroid/nearest source class centroid over training iterations on task $V \rightarrow M$. (c) The homogeneity of each cluster over training iterations on task $V \rightarrow M$. (d) Visualization of target features in the projected space with pseudo-labels. (e) Visualization of target features in the projected space with ground-truth labels.

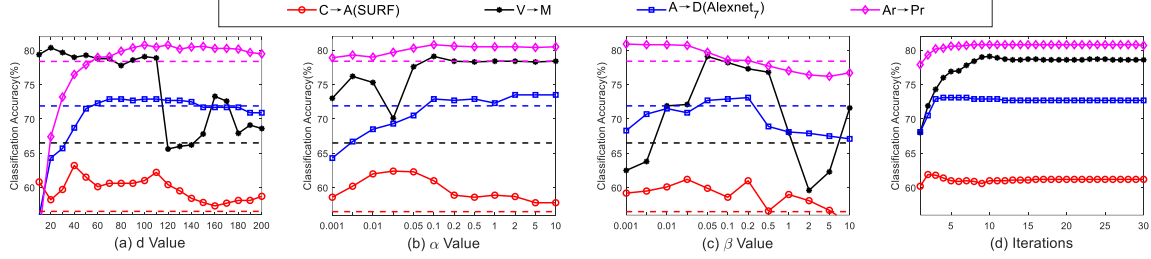


Figure 6. Parameter sensitivity analysis with respect to d , α , β and convergence analysis.

TABLE XI
MEAN OF AVERAGE CLASSIFICATION ACCURACIES (%) FOR DATA
IMBALANCE ISSUE (R: MINIMUM RETENTION RATE).

| r | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|--------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| EasyTL | 32.6 | 34.1 | 36.2 | 38.4 | 39.6 | 40.0 | 42.2 | 42.5 | 43.1 |
| CAPLS | 32.3 | 34.5 | 35.9 | 38.3 | 39.2 | 39.3 | 41.1 | 41.6 | 41.9 |
| MEDA | 32.5 | 34.5 | 37.7 | 40.2 | 41.7 | 42.8 | 44.2 | 45.5 | 46.8 |
| MCS | 32.9 | 36.3 | 39.4 | 40.1 | 41.1 | 41.9 | 43.0 | 42.8 | 44.1 |
| SPL | 34.1 | 37.7 | 38.0 | 40.4 | 40.9 | 42.5 | 43.3 | 43.1 | 45.3 |
| CMMS | 35.6 | 38.0 | 39.5 | 42.3 | 43.6 | 44.7 | 46.8 | 47.8 | 48.0 |

domain as target and the rest as sources by turns, *i.e.*, we have three cross-domain tasks. And then, similar to the setting of [76] to study the performance of clustering algorithm on imbalanced datasets, we randomly sample nine subsets from the new dataset with different minimum retention rates. For the minimum retention rate r , the samples of the first category for target data will be kept with probability r , the last category with probability 1, and the other categories linearly in between. We summarize the results in Table XI, where we report the mean of average classification accuracy on three tasks over five times. We observe that data imbalance issue has negative influence on all methods. However, our CMMS could still obtain the highest accuracy with various minimum retention rates, which illustrates the advantage of our method on imbalanced dataset. One of the potential reasons could be that the local manifold self-learning strategy introduced by our CMMS can capture more inherent structure of target data, thus making the influence of the datasets imbalance reduced. Even so, this insight inspires us to explore better solution of such issue in our future work.

4) *Parameters Sensitivity and Convergence Analysis:* In our CMMS, there are three tunable parameters: α , β and d . We conducted extensive parameter sensitivity analysis with respect to the three parameters in a wide range. We vary one parameter once and fix the others as the optimal values. The results

of $C \rightarrow A$ (SURF), $V \rightarrow M$, $A \rightarrow D$ (Alexnet₇) and $Ar \rightarrow Pr$ are reported in Fig. 6 (a) ~ (c). Meanwhile, to demonstrate the effectiveness of our CMMS, we also display the results of the best competitor as the dashed lines.

First, we investigate the sensitivity of d in a wide range [10, 20, ..., 200]. As depicted in Fig. 6 (a), the optimal value of d varies on different datasets. Nevertheless, our CMMS can consistently achieve better performance within a wide range $d \in [90, 110]$. Note that, all the experimental results of our CMMS reported in this paper are achieved by fixing d to 100, which demonstrates the potential of performance improvements of our approach. Then, we run our CMMS as α varies from 0.001 to 10.0. From Fig. 6 (a), it is observed that when the value of α is much small, it may not be contributed to the improvement of performance. Whereas, with the appropriate increase of α , the clustering process of target data is emphasized, and thus our CMMS can exploit the cluster structure information more effectively. We find that, when α is located within a wide range [0.1, 10.0], our proposal can consistently achieve optimal performance. Next, we evaluate the influence of parameter β on our CMMS by varying the value from 0.001 to 10.0. It is infeasible to determine the optimal value of β , since it highly depends on the domain prior knowledge of the datasets. However, we empirically find that, when β is located within the range [0.05, 0.2], our CMMS can obtain better classification results than the most competitive competitor. Furthermore, we observe that in Fig. 6 (a) ~ (c), the curves of task $V \rightarrow M$ show higher fluctuations compared to other tasks. The potential reason may be that the data imbalance on task $V \rightarrow M$ makes our CMMS more sensitive to the setting of parameter values. Despite this, within the reasonable range of three parameter settings discussed above, our proposal can still significantly outperform the rivals. We also display the convergence analysis in Fig. 6 (d). We can see that CMMS can quickly converge within 10 iterations.

TABLE XII
CLASSIFICATION ACCURACIES (%) ON OFFICE-CALTECH10 DATASET WITH SURF FEATURES FOR SEMI-SUPERVISED DOMAIN ADAPTATION

| Task | SVM _t | SVM _{st} | MMDT | DTMKL-f | CDLS | ILS | TFMKL-S | TFMKL-H | CMMS |
|---------|------------------|-------------------|------|---------|-------------|-------------|-------------|-------------|-------------|
| A→C | 31.1 | 42.4 | 36.4 | 40.5 | 37.1 | 43.6 | 43.8 | 43.5 | 43.0 |
| A→D | 56.9 | 47.7 | 56.7 | 45.9 | 61.9 | 49.8 | 62.0 | 57.3 | 61.8 |
| A→W | 62.8 | 50.1 | 64.6 | 47.9 | 69.3 | 59.7 | 70.9 | 71.4 | 69.2 |
| C→A | 44.7 | 47.2 | 49.4 | 47.3 | 52.5 | 55.1 | 54.2 | 54.4 | 57.7 |
| C→D | 56.3 | 52.0 | 56.5 | 52.2 | 59.8 | 56.2 | 60.1 | 60.4 | 59.1 |
| C→W | 60.0 | 54.5 | 63.8 | 54.4 | 68.7 | 62.9 | 68.1 | 68.6 | 67.2 |
| D→A | 44.7 | 44.3 | 46.9 | 41.6 | 51.8 | 55.0 | 53.1 | 50.8 | 55.2 |
| D→C | 31.3 | 36.8 | 34.1 | 36.0 | 36.9 | 41.0 | 38.9 | 37.9 | 39.5 |
| D→W | 62.0 | 80.6 | 74.1 | 77.6 | 70.7 | 80.1 | 79.1 | 76.7 | 82.5 |
| W→A | 45.4 | 45.2 | 47.7 | 45.3 | 52.3 | 54.3 | 54.4 | 54.0 | 53.4 |
| W→C | 29.7 | 36.1 | 32.2 | 36.3 | 35.1 | 38.6 | 36.2 | 34.9 | 37.7 |
| W→D | 56.5 | 71.2 | 67.0 | 69.6 | 61.3 | 70.8 | 69.1 | 69.3 | 72.9 |
| Average | 48.4 | 50.7 | 52.5 | 49.6 | 54.8 | 55.6 | 57.5 | 56.6 | 58.3 |

TABLE XIII
CLASSIFICATION ACCURACIES (%) ON MSRC-VOC2007 DATASET FOR SEMI-SUPERVISED DOMAIN ADAPTATION

| | Task | SVM _t | SVM _{st} | MMDT | DTMKL-f | CDLS | ILS | TFMKL-S | TFMKL-H | CMMS |
|-----------|---------|------------------|-------------------|------|---------|------|------|-------------|---------|-------------|
| $n_l = 2$ | M→V | 28.9 | 38.5 | 35.1 | 36.8 | 30.2 | 34.2 | 38.2 | 36.6 | 35.8 |
| | V→M | 55.9 | 55.5 | 59.9 | 64.1 | 55.8 | 49.9 | 68.5 | 71.0 | 76.8 |
| | Average | 42.4 | 47.0 | 47.5 | 50.5 | 43.0 | 42.1 | 53.4 | 53.8 | 56.3 |
| $n_l = 4$ | M→V | 30.2 | 39.0 | 36.0 | 36.9 | 31.7 | 35.4 | 38.4 | 36.8 | 36.2 |
| | V→M | 64.4 | 56.6 | 62.1 | 65.0 | 57.3 | 50.2 | 70.4 | 71.4 | 77.7 |
| | Average | 47.3 | 47.8 | 49.1 | 51.0 | 44.5 | 42.8 | 54.4 | 54.1 | 57.0 |

TABLE XIV
CLASSIFICATION ACCURACIES (%) ON OFFICE-CALTECH10 DATASET FOR SURF → DECAF₆

| Task | SVM _t | MMDT | SHFA | CDLS | Li [42] | CDSPP | CMMS |
|---------|------------------|------|------|------|-------------|-------------|-------------|
| A→C | 79.8 | 78.1 | 79.5 | 83.8 | 84.3 | 87.9 | 88.6 |
| A→W | 90.5 | 89.4 | 90.1 | 93.6 | 93.2 | 94.9 | 93.2 |
| C→A | 89.0 | 87.5 | 88.6 | 90.7 | 92.6 | 92.5 | 93.2 |
| C→W | 90.5 | 88.9 | 89.6 | 92.5 | 92.1 | 94.0 | 93.2 |
| W→A | 89.0 | 88.3 | 88.7 | 90.5 | 93.9 | 92.3 | 93.3 |
| W→C | 79.8 | 78.6 | 79.7 | 82.1 | 84.9 | 88.1 | 88.5 |
| Average | 86.4 | 85.1 | 86.0 | 88.9 | 90.2 | 91.6 | 91.7 |

TABLE XV
CLASSIFICATION ACCURACIES (%) ON MULTILINGUAL REUTERS COLLECTION DATASET (SPANISH AS THE TARGET DOMAIN)

| Source | SVM _t | MMDT | SHFA | CDLS | Li [42] | CDSPP | CMMS |
|---------|------------------|------|------|------|---------|-------|-------------|
| English | 67.4 | 67.8 | 68.9 | 70.8 | 71.1 | 69.1 | 74.7 |
| French | | 68.3 | 69.1 | 71.2 | 71.2 | 69.0 | 74.4 |
| German | | 67.7 | 68.3 | 71.0 | 70.9 | 68.8 | 74.7 |
| Italian | | 66.5 | 67.5 | 71.7 | 71.5 | 68.8 | 74.6 |
| Average | 67.4 | 67.6 | 68.5 | 71.2 | 71.2 | 68.9 | 74.6 |

D. Semi-supervised Domain Adaptation

1) *Results in Homogeneous Setting*: The averaged classification results of all methods on the Office-Caltech10 dataset over 20 random splits are shown in Table XII. We can observe that regarding the total average accuracy, our CMMS can obtain 0.8% improvement compared with the second best method TFMKL-S. The results on the MSRC-VOC2007 dataset over 5 random splits are shown in Table XIII. Some results are cited from [24]. Compared with the most comparative competitors, CMMS can achieve 2.5% and 2.6% improvement when the number of labeled target samples in per class is set to 2 and 4, respectively.

2) *Results in Heterogeneous Setting*: The results on the Office-Caltech10 and Multilingual Reuters Collection datasets are listed in Table XIV and Table XV. Some results are cited from [42]. We can observe that our CMMS achieves

the optimal performance in terms of the average accuracy on both datasets. Specifically, compared with the best competitors, 0.1% and 3.4% improvements are obtained. Especially, CMMS works the best for 7 out of all 10 tasks on two datasets, which adequately confirms the excellent generalization capacity of our CMMS in the heterogeneous setting.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a novel domain adaptation method named CMMS is proposed. Unlike most of existing methods that generally assign pseudo-labels to target data independently, CMMS makes label prediction for target samples by the class centroid matching of source and target domains, such that the data distribution structure of target domain can be exploited. To explore the structure information of target data more thoroughly, a local manifold self-learning strategy is further introduced into CMMS, which can capture the inherent local manifold structure of target data by adaptively learning the data similarity in the projected space. The CMMS optimization problem is not convex with all variables, and thus an iterative optimization algorithm is designed to solve it, whose computational complexity and convergence are carefully analyzed. We further extend CMMS to the semi-supervised scenario including both homogeneous and heterogeneous settings which are appealing and promising. Extensive experimental results on seven datasets reveal that CMMS significantly outperforms the baselines and several state-of-the-art methods in both unsupervised and semi-supervised scenarios.

Future research will include the following: 1) Considering the computational bottleneck of CMMS optimization, we will design more efficient algorithm for the local manifold self-learning; 2) Except for the class centroids, we can introduce additional measure to represent the structure of data distribution, such as the covariance; 3) In this paper, we extend CMMS to the semi-supervised scenario in a direct but effective way. In

the future, more elaborate design of semi-supervised methods is worth further exploration; 4) It would be a promising research direction to design a systematical way to determine the optimal value of d .

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their constructive comments which helped to improve the quality of this article. The authors are thankful for the financial support by the the Key-Area Research and Development Program of Guangdong Province 2019B010153002 and the National Natural Science Foundation of China (U1636220, 61961160707 and 61772525).

REFERENCES

- [1] 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.
- [2] Z. Guo and Z. Wang, "Cross-domain object recognition via input-output kernel analysis," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3108-3119, Aug. 2013.
- [3] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801-814, Apr. 2019.
- [4] C.-X. Ren, D.-Q. Dai, K.-K. Huang, and Z.-R. Lai, "Transfer learning of structured representation for face recognition," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5440-5454, Dec. 2014.
- [5] Q. Qiu and R. Chellappa, "Compositional dictionaries for domain adaptive face recognition," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5152-5165, Dec. 2015.
- [6] A. J. Ma, J. Li, P. C. Yuen, and P. Li, "Cross-domain person reidentification using domain adaptation ranking SVMs," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1599-1613, May 2015.
- [7] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076-1089, May 2014.
- [8] 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.
- [9] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2013, pp. 2200-2207.
- [10] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066-2073.
- [11] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. Int. Conf. Comput. Vis.*, Aug. 2013, pp. 2960-2967.
- [12] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. Amer. Assoc. Artif. Intell. Conf.*, 2016, pp. 2058-2065.
- [13] T. Yao, Y. Pan, Ch.-W. Ngo, H. Li, and T. Mei, "Semi-supervised domain adaptation with subspace learning for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2142-2150.
- [14] A. Gretton, K. M. Borgwardt, M. Rasch, B. Scholkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. in Neural Inf. Process. Syst.*, 2007, pp. 513-520.
- [15] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, 1967, pp. 281-297.
- [16] S. Li, S. Song, G. Huang, and Z. Ding, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4260-4273, Sept. 2018.
- [17] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1-6.
- [18] A. Goh and R. Vidal, "Clustering and dimensionality reduction on Riemannian manifolds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1-7.
- [19] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119-155, Dec. 2003.
- [20] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2001, pp. 585-591.
- [21] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 977-986.
- [22] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko, "Asymmetric and category invariant feature transformations for domain adaptation," *Int. J. Comput. Vis.*, vol. 41, nos. 1-2, pp. 28-41, 2014.
- [23] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3845-3854.
- [24] W. Wang, H. Wang, Z. X. Zhang, C. Zhang, and Y. Gao, "Semi-supervised domain adaptation via Fredholm integral based kernel methods," *Pattern Recognit.*, vol. 85, pp. 185-197, Jan. 2019.
- [25] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134-1148, Jun. 2014.
- [26] Y.-T. Hsieh, S.-Y. Tao, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Recognizing heterogeneous cross-domain data via generalized joint distribution adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2016, pp. 1-6.
- [27] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1410-1417.
- [28] S. Chen, F. Zhou, and Q. Liao, "Visual domain adaptation using weighted subspace alignment," in *Proc. SPIE Int. Conf. Vis. Commun. Image Process.*, Nov. 2016, pp. 1-4.
- [29] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. ACM Int. Conf. Multimedia*, Oct. 2018, pp. 402-410.
- [30] L. Zhang. (2019). "Transfer adaptation learning: a decade survey." [Online]. Available: <https://arxiv.xileisou.top/abs/1903.04687>
- [31] J. Liang, R. He, 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.
- [32] J. Liang, R. He, Z. Sun and T. Tan, "Distant Supervised Centroid Shift: A Simple and Efficient Approach to Visual Domain Adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2975-2984.
- [33] Q. Wang and T. P. Breckon. (2019). "Unsupervised domain adaptation via structured prediction based selective pseudo-labeling." [Online]. Available: <https://arxiv.xileisou.top/abs/1911.07982>
- [34] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 97-105.
- [35] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136-144.
- [36] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 1180-1189.
- [37] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 7167-7176.
- [38] A. Chadha, and Y. Andreopoulos, "Improved techniques for adversarial discriminative domain adaptation," *IEEE Trans. Image Process.*, vol. 29, pp. 2622-2637, Nov. 2019.
- [39] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8004-8013.
- [40] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 627-636.
- [41] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2239-2247.
- [42] J. Li, K. Lu, Z. Huang, L. Zhu, and H. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381-1391, May 2019.
- [43] Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5081-5090.
- [44] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54-66, Jan. 2015.

- [45] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei, "Semi-supervised domain adaptation with subspace learning for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2142-2150.
- [46] D. Hong, N. Yokoya, and X. Zhu, "Learning a robust local manifold representation for hyperspectral dimensionality reduction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2960-2975, Jun. 2017.
- [47] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261-1270, Mar. 2019.
- [48] C. Hou, F. Nie, H. Tao, and D. Yi, "Multi-view unsupervised feature selection with adaptive similarity and view weight," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1998-2011, Sept. 2017.
- [49] W. Wang, Y. Yan, F. Nie, S. Yan, and N. Sebe, "Flexible manifold learning with optimal graph for image and video representation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2664-2675, Jun. 2018.
- [50] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5552-5562, Dec. 2016.
- [51] J. Li, M. Jing, K. Lu, L. Zhu and H. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6103-6115, Dec. 2019.
- [52] S. Wang, J. Lu, X. Gu, H. Du, and J. Yang, "Semi-supervised linear discriminant analysis for dimension reduction and classification," *Pattern Recognit.*, vol. 57, pp. 179-189, Sept. 2016.
- [53] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213-226.
- [54] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647-655.
- [55] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 5018-5027.
- [56] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. (2017). "VisDA: the visual domain adaptation challenge." [Online]. Available: <https://arxiv.org/abs/1710.06924>
- [57] M.-R. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views-an application to multilingual text categorization," in *Proc. Adv. in Neural Inf. Process. Syst.*, Dec. 2009, pp. 28-36.
- [58] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1859-1867.
- [59] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang, "Easy transfer learning by exploiting intra-domain structures," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2019, pp. 1210-1215.
- [60] Q. Wang, P. Bu, and T. P. Breckon, "Unifying unsupervised domain adaptation and zero-shot visual recognition," in *Proc. Int. Joint Conf. Neural Netw.*, Sep. 2019, pp. 14-19.
- [61] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proc. Amer. Assoc. Artif. Intell. Conf.*, 2019, pp. 5345-5352.
- [62] V. K. Kurmi, S. Kumar, and V. P. Nambodiri, "Attending to discriminative certainty for domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 491-500.
- [63] S. Li, C. Liu, Q. Lin, B. Xie, Z. Ding, G. Huang, and J. Tang, "Domain conditioned adaptation network," in *Proc. Amer. Assoc. Artif. Intell. Conf.*, 2020, pp. 11386-11393.
- [64] Y.-W. Luo, C.-X. Ren, P. Ge, K.-K. Huang, and Y.-F. Yu. (2020). "Unsupervised domain adaptation via discriminative manifold embedding and alignment." [Online]. Available: <https://arxiv.org/abs/2002.08675>
- [65] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, and Q. Tian, "Gradually vanishing bridge for adversarial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12455-12464.
- [66] X. Gu, J. Sun, and Z. Xu, "Spherical space domain adaptation with robust pseudo-label loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9101-9110.
- [67] Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong and Q. He, "Deep subdomain adaptation network for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1-10, May 2020.
- [68] S. Wu, J. Zhong, W. Cao, R. Li, Z. Yu, and H.-S. Wong, "Improving domain-specific classification by collaborative learning with adaptation networks," in *Proc. Amer. Assoc. Artif. Intell. Conf.*, 2019, pp. 5450-5457.
- [69] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and Z. Huang, "Cycle-consistent conditional adversarial transfer networks," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, Oct. 2019, pp. 747-755.
- [70] S. Lee, D. Kim, N. Kim, and S.-G. Jeong, "Drop to adapt: learning discriminative features for unsupervised domain adaptation," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2019, pp. 91-100.
- [71] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. (2019). "Adversarial domain adaptation with domain mixup." [Online]. Available: <https://arxiv.org/abs/1912.01805>
- [72] M. Chen, S. Zhao, H. Liu, and D. Cai, "Adversarial-learned loss for domain adaptation," in *Proc. Amer. Assoc. Artif. Intell. Conf.*, 2020, pp. 3521-3528.
- [73] 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.
- [74] Q. Wang and T. P. Breckon. (2020). "Cross-domain structure preserving projection for heterogeneous domain adaptation." [Online]. Available: <https://arxiv.org/abs/2004.12427>
- [75] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579-2605, Nov. 2008.
- [76] J. Chang, G. Meng, L. Wang, S. Xiang and C. Pan, "Deep self-evolution clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 809-823, Dec. 2018.