Domain Invariant and Class Discriminative Feature Learning for Visual Domain Adaptation

Shuang Li, Shiji Song, Gao Huang, Zhengming Ding, *Student Member, IEEE*, and Cheng Wu

***Abstract*— Domain adaptation manages to build an effective target classifier or regression model for unlabeled target data by utilizing the well-labeled source data but lying different distributions. Intuitively, to address domain shift problem, it is crucial to learn domain invariant features across domains, and most existing approaches have concentrated on it. However, they often do not directly constrain the learned features to be class discriminative for both source and target data, which is of vital importance for the final classification. Therefore, in this paper, we put forward a novel feature learning method for domain adaptation to construct both *domain invariant and class discriminative* representations, referred to as DICD. Specifically, DICD is to learn a latent feature space with important data properties preserved, which reduces the domain difference by jointly matching the marginal and class-conditional distributions of both domains, and simultaneously maximizes the inter-class dispersion and minimizes the intra-class scatter as much as possible. Experiments in this paper have demonstrated that the class discriminative properties will dramatically alleviate the cross-domain distribution inconsistency, which further boosts the classification performance. Moreover, we show that exploring both domain invariance and class discriminativeness of the learned representations can be integrated into one optimization framework, and the optimal solution can be derived effectively by solving a generalized eigen-decomposition problem. Comprehensive experiments on several visual cross-domain classification tasks verify that DICD can outperform the competitors significantly.**

***Index Terms*— Domain adaptation, feature extraction, subspace learning.**

I. Introduction

IN REAL-WORLD visual pattern recognition fields, labeled information plays a fundamental role to construct an effective classification or regression model. The reliance on abundant labeled data can however constitute a limitation, since labeling instances is often time-consuming or expensive to come by for new visual domains. One may expect to leverage available well-labeled data in a related source domain to transfer effective knowledge to an unlabeled target domain, and this can provide a promising solution to scenarios that suffer from a shortage of labeled data [1]–[6].

Domain adaptation concerns with the problem of how to accurately adapt classifiers or regression models from a labeled source domain to a different but related unlabeled target domain [1], [7]. In cross-domain problems, data in the source and target domains usually follow diverse distributions, and deducing a good representation shared by involved domains is crucial. A very fruitful line of prior works have focused on how to discover new latent feature representations to reduce the domain difference [2], [8]–[12]. Another line of works aim to build an effective transfer distance metric [13], [14], which can be viewed as a global linear transformation of the input space [15]. Under the learned metric, the distribution difference across two domains can be resolved. Most of existing methods only consider the deduced features or metrics are domain invariant across domains, which will mitigate the domain shift effectively. However, the learned domain invariant features may not only draw both domains close, but also mix all the samples with different classes together, and this will lead to the degradation of the classification performance. Therefore, the discriminative and generalized ability of the learned features are also of great importance for identifying the unlabeled target domain data, as shown in Figure 1.

In this paper, we attempt to address the domain adaptation problem by learning a low-dimensional latent feature space where the projected features across domains are both *Domain Invariant and Class Discriminative*, so we call our method as DICD. The motivation of DICD is illustrated in Figure 1. To be specific, targeting at mitigating the tremendous difference between source and target domains, DICD matches both marginal and conditional distributions effectively measured by Maximum Mean Discrepancy (MMD) [16] in an iterative refine manner. Simultaneously, the learned features need to be not only separable but also discriminative and generalized enough. To this end, DICD explicitly minimizes the distance between every two projected samples with the same label, and maximizes the distance between every pair of projected samples in different classes. This can compact intra-class variations and enlarge inter-class difference, which encourages

Manuscript received April 7, 2017; revised October 12, 2017 and April 1, 2018; accepted May 12, 2018. Date of publication May 22, 2018; date of current version June 1, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1200203 and in part by the National Natural Science Foundation of China under Grants 41427806 and 61273233. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gang Hua. *(Corresponding author: Shiji Song.)*

S. Li, S. Song, and C. Wu are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: l-s12@mails.tsinghua.edu.cn; shijis@mail.tsinghua.edu.cn; wuc@tsinghua.edu.cn).

G. Huang is with the Department of Computer Science, Cornell University, Ithaca, NY 14850 USA (e-mail: gh349@cornell.edu).

Z. Ding is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: allanding@ece.neu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2839528

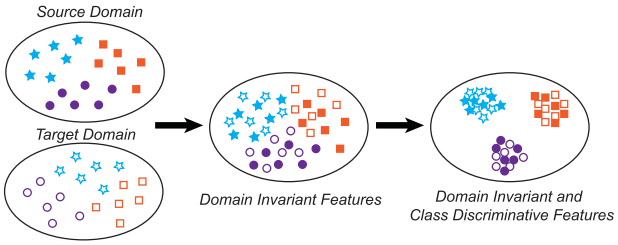


Fig. 1. Illustrating the motivation of the proposed method. Most related works mainly concentrate on learning domain invariant features in a latent subspace to mitigate the divergence across the source and target domains. Our goal is to learn both domain invariant and class discriminative features, which will further boost the recognition performance. (Best viewed in color.)

samples with the same label to form a more compact cluster, and different clusters with different labels are far away.

We believe that the domain invariant and class discriminative properties are complementary to each other. Domain invariance of the learned representations could help target leverage useful labeled source information directly, which paves the way to explore more class-discriminatory information. In addition, class discriminativeness will keep different classes far away with each other, and significantly diminish between-domain distribution discrepancy. The good class-separability features will also facilitate the final prediction.

DICD could integrate the optimization of learning domain invariant and class discriminative features, with data global covariance structure preserved, into a unified framework. On one hand, domain invariant features will adapt the trained source model to target domain successfully. On the other hand, class discriminative features will yield good classification performance by standard classification methods, e.g. k-nearest neighbor (k-NN) classifier [17] and support vector machine (SVM) [18]–[20]. We summarize our contributions as follows.

- DICD aims to learn both domain invariant and class discriminative feature representations for the source and target data, which could effectively minimize the considerable divergence between two domains and fully exploit the discriminative information of both domains to boost the classification performance. DICD explicitly attempts to model the difference between the classes of data while maintaining the domain-invariant information. However, most of domain adaptation methods on the other hand do not take into account any discriminative knowledge in class.
- DICD could effectively integrate learning domain invariant and class discriminative representations into one framework, and obtain the global optimal solution by solving a generalized eigen-decomposition problem. As a result, standard machine learning approaches can be easily applied on the newly learned features of the labeled source data for use in the unlabeled target data.
- Analytical experiments in this paper demonstrate that exploring the class discriminativeness of learned representations can further help mitigate the domain shift. This phenomenon manifests that the domain invariant and class discriminative properties are complementary to each other, and the discriminative information is of vital importance to construct robust and perfect cross-domain feature representations.
- Comprehensive experiments on several visual datasets: CMU-PIE, COIL20, USPS, MNIST, Office, Caltech256, demonstrate that DICD is superior to other state-of-the-arts domain adaptation methods with a large margin.

The rest of paper is organized as follows. We first briefly review related works in Section II. In Section III, our method DICD will be proposed to learn both domain invariant and class discriminative features for domain adaptation problems. In Section IV, empirical studies on several real-world visual datasets and analytical experiments will be conducted. Finally, we conclude this paper in Section V.

II. Related Works

There has been extensive prior work on domain adaptation, and according to the related work, one can roughly divide existing domain adaptation approaches into three categories: instance reweighting [21]–[25], feature extraction [2], [8]–[12], [26] and classifier adaptive approaches [27]–[29].

Most instance reweighting methods reweight source data [21], [22] to reduce the distribution difference across domains. Importance weighted cross validation (IWCV) proposed in [22] is to alleviate misestimation due to the difference between source and target by reweighting the source data. It needs estimate the source and target densities accurately to calculate the reweighting coefficients. Huang *et al.* [21] propose a Kernel Mean Matching (KMM) method to close the means of both domains in a reproducing kernel Hilbert space, which can compute the source data weights directly without estimating the biased densities or selection probabilities. However, different from the proposed method, IWCV and KMM assume the conditional distributions across domains to be unchanged, which is often violated in real-world applications. DICD will match both marginal and conditional distributions to mitigate the domain discrepancy across source and target. Since DICD is of particular relevance to feature extraction approaches, we will focus on them to discuss.

As we all know, feature extraction is crucial for various image applications. In order to capture the knowledge of both labeled tags and related concept of images for representation learning, Cui *et al.* [30] propose a relational regularized regression CNN (R^3CNN), which is a knowledge base embedded image representation learning approach. But R^3CNN is not designed to address domain adaptation problems. Shen *et al.* [31] introduce causality into image classification under the non-i.i.d situation, and present a causally regularized logistic regression (CRLR) algorithm. Pan *et al.* [12] put forward a transfer component analysis (TCA) approach, to decrease the distance between source and target domains via the learned transfer components underlying the both domains. Furthermore, joint distribution adaptation (JDA), proposed in [10], reduces the difference across two domains by jointly adapting their marginal and conditional distributions in a principal dimension reduction procedure. Transfer joint

matching (TJM) [11] aims to mitigate the domain shift by matching features in the latent space and reweighting source instances simultaneously.

These approaches mainly concentrate on deriving domain invariant feature representations to enable transferring classifiers from the source domain to the target domain. However, the perfect learned features should not only minimize the distribution discrepancy across domains effectively, but also be discriminative enough and benefit the final classification. DICD can achieve these two goals simultaneously and yield good classification performance.

Based on JDA, [32] proposes to exploit the local structure information of unlabeled target data to improve the target prediction accuracy in the second stage. A domain invariant projection (DIP) method proposed in [8] focuses on matching both source and target marginal distributions in a low-dimensional latent space, rather than in the original feature space. DIP also explores the label information of source data by encouraging the projected source data to form clusters in the latent space.

However, DICD aims to match both marginal and conditional distributions of the source and target domains, which is more effective to reduce the domain shift. Moreover, DICD can fully explore and utilize the discriminative knowledge of target data to discover the underlying information of target domain to boost the final classification accuracy.

Another related literature work is transfer metric learning approache, which aims to learn a transferable distance metric scheme between involved domains [13], [14]. Geng *et al.* propose a domain adaptation metric learning (DAML) approach, which introduces a data dependent MMD regularization term to the traditional metric learning method. The work in [14] develops a robust transfer metric learning (RTML) framework to mitigate the domain discrepancy in both sample space and feature space. Whereas DICD obtains a projection matrix to transform original data to a low-dimensional feature space, in which the domain difference between two domains can be effectively reduced as well as the discriminative information can be extracted.

To our knowledge, DICD is among the very leading approaches to derive both domain invariant and class discriminative feature representations for domain adaptation. This strategy will lead DICD to achieve better recognition accuracies than competitive approaches with a significant advantage.

III. Proposed Approach

In this section, we will start from the problem definition and then present our approach to learn both domain invariant and class discriminative (DICD) feature representations for visual domain adaptation. Notably, in this paper, all the frequently used notations are summarized in Table I.

A. Preliminary & Motivation

In domain adaptation problems, labeled data $\mathcal{D}_S$ are available in the *source domain*, and there are only unlabeled data $\mathcal{D}_T$ in the *target domain*. We denote $\mathcal{D}_S = \{(\boldsymbol{x}_{Si}, y_{Si})\}_{i=1}^{n_s} = \{\mathbf{X}_S, \boldsymbol{y}_S\}$, and $\mathcal{D}_T = \{(\boldsymbol{x}_{Tj}\}_{j=1}^{n_t} = \{\mathbf{X}_T\}$, where $\boldsymbol{x}_{Si}, \boldsymbol{x}_{Tj} \in \mathcal{X}$ ($\mathcal{X} \subset \mathbb{R}^m$), and $y_{Si} \in \mathcal{Y}$ ($\mathcal{Y} \subset \mathbb{R}$) is the corresponding label of the source data $\boldsymbol{x}_{Si}$. n_s, n_t are the numbers of the source and target samples, and $\mathcal{X}$, $\mathcal{Y}$ are the data input space and label space respectively.

TABLE I
Frequently Used Notations in This Paper

Notation	Description	Notation	Description
$\mathcal{D}_S/\mathcal{D}_T$	source/target domain data	$\mathbf{X}_S/\mathbf{X}_T$	source/target data matrix
n_s/n_t	#source/target samples	$\mathbf{X}$	original data matrix
m	#original features	$\mathbf{Z}$	embedding data matrix
d	#projected features	$\mathbf{P}$	projected matrix
C	#shared classes	$\mathbf{H}$	centering matrix
N	#iterations	$\mathbf{W}$	MMD matrix
β	regularization parameter	$\mathbf{D}_{same}/$ $\mathbf{D}_{diff}$	intra-class/inter-class distance calculation matrix
α,ρ	trade-off parameters	$\mathcal{L}_{same}/$ $\mathcal{L}_{diff}$	intra-class/inter-class distance loss term

Actually, a majority of domain adaptation approaches focus on reducing the distribution distance between two domains using different kinds of metrics (e.g. K-L divergence [33] or MMD [10], [12], [34]). However, we expect the transformed feature representations to combine (i) domain-invariance and (ii) discriminativeness.

1) Domain-Invariance: To benefit transferring knowledge from the source domain to the target domain, the learned representations of both domains in the latent subspace should be well aligned. Then standard classification methods could be trained in the source domain and fit well for the target data. Therefore, we propose to match both marginal and conditional distributions across domains by minimizing their empirical MMD distance. However, if we only focus on reducing the domain shift, the derived features may be tortured or squeezed to minimize the distance metric, which will lose the intrinsic discriminative knowledge.

2) Discriminativeness: Since the discriminative information is of vital importance for the final classification, we need the projected features to be not only invariant across domains, but also class discriminative (especially for the target data), which encourages samples in the same class to be close enough to form a compact cluster, and different class clusters to be far away. To this end, we enforce to minimize the distance of every two samples in the same class, and maximize the distance of every two samples sharing different labels, in the source and target domains respectively. Moreover, experiments in Section IV demonstrate that the discriminativeness of the learned features will help align the same class samples in source and target more effectively.

Then we will present the proposed approach from these two perspectives.

B. Learning Domain Invariant Features

1) Domain-Wise Adaptation: When $\mathbf{X}_S$ and $\mathbf{X}_T$ are drawn from different domains, it is very essential to minimize the marginal distribution difference across two domains by learning an effective cross-domain metric. This issue is of particular importance and gains its popularity in transfer learning. Lots of recent research activities adopt the criterion Maximum Mean Discrepancy (MMD) to measure the distribution distance

between two domains, that is, the means of two domains tend to be pulled close together.

To this end, we propose *domain-wise adaptation* to guide the metric learning by leveraging mean discrepancy of two domains as follows:

$$\begin{aligned}\mathcal{L}_{MMD}^{(0)} &= \left\| \frac{1}{n_s}\sum_{i=1}^{n_s} \boldsymbol{z}_{Si} - \frac{1}{n_t}\sum_{j=1}^{n_t} \boldsymbol{z}_{Tj} \right\|^2 \\ &= \left\| \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{P}^\top \boldsymbol{x}_{Si} - \frac{1}{n_t}\sum_{j=1}^{n_t} \mathbf{P}^\top \boldsymbol{x}_{Tj} \right\|^2 \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}\mathbf{W}_0\mathbf{X}^\top\mathbf{P}\right), \end{aligned} \tag{1}$$

where $\mathbf{X} = [\mathbf{X}_S, \mathbf{X}_T]$, and $\boldsymbol{z}_{Si}, \boldsymbol{z}_{Tj}$ are the transformed low-dimensional representations of the source and target data. The loss term $\mathcal{L}_{MMD}^{(0)}$ can measure the distance between data means of two domains in the latent subspace, and $\mathbf{W}_0$ is the marginal MMD matrix [10]. We denote $\mathbf{1}_{m\times n}$ as an m by n matrix whose elements are all one and $\mathbf{0}_{m\times n}$ as zero matrix with elements being all zero, then we can calculate $\mathbf{W}_0$ as

$$\mathbf{W}_0 = \begin{bmatrix} \frac{1}{n_s^2}\mathbf{1}_{n_s\times n_s} & -\frac{1}{n_s n_t}\mathbf{1}_{n_s\times n_t} \\ -\frac{1}{n_s n_t}\mathbf{1}_{n_t\times n_s} & \frac{1}{n_t^2}\mathbf{1}_{n_t\times n_t} \end{bmatrix}. \tag{2}$$

However, reducing the difference in the marginal distributions cannot guarantee that the conditional distributions across domains are also be drawn close. Indeed, minimizing the difference between the conditional distributions is crucial for effective distribution adaptation. Unfortunately, it is non-trivial to match the conditional distributions, even by exploring sufficient statistics of the distributions, since there is no labeled data in the target domain. Some very recent works started to match the conditional distributions via circular validation and co-training. However, they cannot address the problem where there is no labeled data in the target domain.

2) Class-Wise Adaptation: To address this problem, we propose to explore the pseudo labels of the target data, which can be easily predicted by applying some basic classifiers trained on the labeled source data to the unlabeled target data [10], [14]. This strategy aims to uncover the underlying structure of two domains by transferring the local information. Now with the true source labels and pseudo target labels, we can essentially match the class conditional distributions and propose *class-wise adaptation* as follows:

$$\begin{aligned}\mathcal{L}_{MMD}^{(k)} &= \left\| \frac{1}{n_s^{(k)}} \sum_{\boldsymbol{x}_{Si}\in\mathcal{D}_S^{(k)}} \boldsymbol{z}_{S_i}^{(k)} - \frac{1}{n_t^{(k)}} \sum_{\boldsymbol{x}_{Ti}\in\hat{\mathcal{D}}_T^{(k)}} \boldsymbol{z}_{T_j}^{(k)} \right\|^2 \\ &= \left\| \frac{1}{n_s^{(k)}} \sum_{\boldsymbol{x}_{Si}\in\mathcal{D}_S^{(k)}} \mathbf{P}^\top \boldsymbol{x}_{S_i}^{(k)} - \frac{1}{n_t^{(k)}} \sum_{\boldsymbol{x}_{Ti}\in\hat{\mathcal{D}}_T^{(k)}} \mathbf{P}^\top \boldsymbol{x}_{T_j}^{(k)} \right\|^2 \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}\mathbf{W}_k\mathbf{X}^\top\mathbf{P}\right), \end{aligned} \tag{3}$$

where $\mathcal{D}_S^{(k)}$ denotes all the source instances with their true class labels being k, and $\boldsymbol{x}_S{}^{(k)} \in \mathcal{D}_S^{(k)}$. $\boldsymbol{z}_S{}^{(k)}$ represents the corresponding transformed source data, and $n_s^{(k)}$ is the number of source instances in class k. Similar definitions apply for the target data according to their pseudo label $\hat{y_T}$. $\mathbf{W}_k$ is the class-conditional MMD matrix for class k, which can be calculated as

$$(\mathbf{W}_k)_{ij} = \begin{cases} \frac{1}{(n_s^{(k)})^2}, & \text{if } \boldsymbol{x}_i,\ \boldsymbol{x}_j \in \mathcal{D}_S^{(k)}; \\ -\frac{1}{n_s^{(k)} n_t^{(k)}}, & \text{if } \boldsymbol{x}_i \in \mathcal{D}_S^{(k)},\ \boldsymbol{x}_j \in \hat{\mathcal{D}}_T^{(k)}; \\ -\frac{1}{n_s^{(k)} n_t^{(k)}}, & \text{if } \boldsymbol{x}_i \in \hat{\mathcal{D}}_T^{(k)},\ \boldsymbol{x}_j \in \mathcal{D}_S^{(k)}; \\ \frac{1}{(n_t^{(k)})^2}, & \text{if } \boldsymbol{x}_i,\ \boldsymbol{x}_j \in \hat{\mathcal{D}}_T^{(k)}; \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We can rewrite the whole MMD loss term as

$$\begin{aligned}\mathcal{L}_{MMD} &= \sum_{k=0}^{C} \mathcal{L}_{MMD}^{(k)} = \sum_{k=0}^{C} \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}\mathbf{W}_k\mathbf{X}^\top\mathbf{P}\right) \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}\mathbf{W}\mathbf{X}^\top\mathbf{P}\right). \end{aligned} \tag{5}$$

We define $\mathbf{W} = \sum_{k=0}^{C} \mathbf{W}_k$, and minimizing (5) will effectively reduce the difference between the source and target domains.

Here, we utilize target pseudo labels to calculate the MMD distance of class-conditional distributions across domains. In the beginning, some of the target pseudo labels maybe incorrect, but we can refine the pseudo labels at an iterative manner. Specifically, we can learn a new source classifier in the low-dimensional common space iteratively, and predict the target data more accurately, since in the learned latent subspace, the distributions of source and target will get closer after every iteration. We will justify this in our experiments.

It is worth noting that target pseudo labels can provide us a new perspective to discover the underlying information of the target domain. We will explore more about target pseudo labels to help us learn class discriminative features.

C. Learning Class Discriminative Features

Besides minimizing the distance between two domains, DICD also expects the projected matrix $\mathbf{P}$ to make the transformed low-dimensional representations $\boldsymbol{z}_i = \mathbf{P}^T\boldsymbol{x}_i$ preserve the intrinsic data structure of the original data as well as the label cluster constraints, i.e. data belonging to the same class should be close while data involved in the different classes should be far in the learned low-dimensional space.

The intuition of learning class discriminative features is derived by minimizing a loss term that consists of two parts. The first part minimizes the distance between instances in the same class, while the second part penalizes all the distance between instances with non-matching labels.

1) Minimizing Intra-Class Compactness: To develop an effective loss term to improve the discriminative power of the learned features, we first expect to reduce the intra-class variation by minimizing the distance between the same class instances in the source and target domains, respectively.

Specifically, for the source data, we can formulate the distance as

$$\begin{aligned}\mathcal{L}_{same}^{(S)} &= \sum_{k=1}^{C} \frac{n_s}{n_s^{(k)}} \sum_{y_{Si}, y_{Sj}=k} \left\| z_{Si} - z_{Sj} \right\|^2 \\ &= \sum_{k=1}^{C} \frac{n_s}{n_s^{(k)}} \sum_{y_{Si}, y_{Sj}=k} \left\| \mathbf{P}^\top \boldsymbol{x}_{Si} - \mathbf{P}^\top \boldsymbol{x}_{Sj} \right\|^2 \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}_S \mathbf{D}_{same}^{(S)} \mathbf{X}_S^\top \mathbf{P} \right), \end{aligned} \tag{6}$$

where

$$\left(\mathbf{D}_{same}^{(S)} \right)_{ij} = \begin{cases} n_s, & \text{if } i = j; \\ -\dfrac{n_s}{n_s^{(k)}}, & \text{if } i \neq j,\ y_{Si} = y_{Sj} = k; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Notably, as [35]–[37], to balance the effects of different classes, we associate different penalty coefficient $\frac{n_s}{n_s^{(k)}}$ on the distances with respect to data from different classes. Since we found if (6) does not have this coefficient for each class, the classes with significantly enough training samples will play a more important role than other classes, and the effects of the class which has less data will be ignored. This usually leads to performance degradation on the target domain. Thus, we aim to alleviate the problem when the samples in each class of the source and target domains are imbalanced. Also, we will explore this in the experiment in Section IV.

Similar to (7), we can obtain the distance term for the target domain using pseudo labels of target data,

$$\begin{aligned}\mathcal{L}_{same}^{(T)} &= \sum_{k=1}^{C} \frac{n_t}{n_t^{(k)}} \sum_{\hat{y_{Ti}}, \hat{y_{Tj}}=k} \left\| z_{Ti} - z_{Tj} \right\|^2 \\ &= \sum_{k=1}^{C} \frac{n_t}{n_t^{(k)}} \sum_{\hat{y_{Ti}}, \hat{y_{Tj}}=k} \left\| \mathbf{P}^\top \boldsymbol{x}_{Ti} - \mathbf{P}^\top \boldsymbol{x}_{Tj} \right\|^2 \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}_T \mathbf{D}_{same}^{(T)} \mathbf{X}_T^\top \mathbf{P} \right), \end{aligned} \tag{8}$$

where

$$\left(\mathbf{D}_{same}^{(T)} \right)_{ij} = \begin{cases} n_t, & \text{if } i = j; \\ -\dfrac{n_t}{n_t^{(k)}}, & \text{if } i \neq j,\ \hat{y_{Ti}} = \hat{y_{Tj}} = k; \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

If we denote $\mathbf{D}_{same} = \mathrm{diag}\left(\mathbf{D}_{same}^{(S)}, \mathbf{D}_{same}^{(T)} \right)$, we can rewrite the intra-class distance loss term of both source and target in a whole formulation as

$$\begin{aligned}\mathcal{L}_{same} &= \mathcal{L}_{same}^{(S)} + \mathcal{L}_{same}^{(T)} \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X} \mathbf{D}_{same} \mathbf{X}^\top \mathbf{P} \right). \end{aligned} \tag{10}$$

Clearly, by minimizing (10), the distance between data in the same class can be explicitly decreased, and the instances with the same label can form compact clusters. But if we expect the clusters are discriminative enough, the inter-class dispersion should be maximized as much as possible.

2) Maximizing Inter-Class Dispersion: We propose to let the distance between instances with non-matching labels in the latent space as large as possible, which will pull different clusters away to improve the discriminativeness of the learned feature representations. Taking the source domain as an example, we can calculate the distance between data in different classes as:

$$\begin{aligned}\mathcal{L}_{diff}^{(S)} &= \sum_{y_{Si} \neq y_{Sj}} \left\| z_{Si} - z_{Sj} \right\|^2 \\ &= \sum_{y_{Si} \neq y_{Sj}} \left\| \mathbf{P}^\top \boldsymbol{x}_{Si} - \mathbf{P}^\top \boldsymbol{x}_{Sj} \right\|^2 \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}_S \mathbf{D}_{diff}^{(S)} \mathbf{X}_S^\top \mathbf{P} \right), \end{aligned} \tag{11}$$

where

$$\left(\mathbf{D}_{diff}^{(S)} \right)_{ij} = \begin{cases} n_s - n_s^{(k)}, & \text{if } i = j,\ y_{Si} = k; \\ -1, & \text{if } i \neq j,\ y_{Si} \neq y_{Sj}; \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Here we need not put extra penalty coefficients for different classes in $\mathcal{L}_{diff}^{(S)}$, since the scale of distance summation between every two instances sharing different labels for each class is at a roughly same level. A similar computing of the distance $\mathcal{L}_{diff}^{(T)}$ for the target domain is

$$\begin{aligned}\mathcal{L}_{diff}^{(T)} &= \sum_{\hat{y_{Ti}} \neq \hat{y_{Tj}}} \left\| z_{Ti} - z_{Tj} \right\|^2 \\ &= \sum_{\hat{y_{Ti}} \neq \hat{y_{Tj}}} \left\| \mathbf{P}^\top \boldsymbol{x}_{Ti} - \mathbf{P}^\top \boldsymbol{x}_{Tj} \right\|^2 \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X}_T \mathbf{D}_{diff}^{(T)} \mathbf{X}_T^\top \mathbf{P} \right), \end{aligned} \tag{13}$$

where

$$\left(\mathbf{D}_{diff}^{(T)} \right)_{ij} = \begin{cases} n_t - n_t^{(k)}, & \text{if } i = j,\ \hat{y_{Ti}} = k; \\ -1, & \text{if } i \neq j,\ \hat{y_{Ti}} \neq \hat{y_{Tj}}; \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

$\mathcal{L}_{diff}^{(S)}$ and $\mathcal{L}_{diff}^{(T)}$ can also be integrated as a whole term, when we denote $\mathbf{D}_{diff} = \mathrm{diag}\left(\mathbf{D}_{diff}^{(S)}, \mathbf{D}_{diff}^{(T)} \right)$. We can express $\mathcal{L}_{diff}$ as

$$\begin{aligned}\mathcal{L}_{diff} &= \mathcal{L}_{diff}^{(S)} + \mathcal{L}_{diff}^{(T)} \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X} \mathbf{D}_{diff} \mathbf{X}^\top \mathbf{P} \right). \end{aligned} \tag{15}$$

In DICD, to achieve class discriminative features in the latent subspace, we should jointly minimize the intra-class variation and maximize the inter-class dispersion explicitly. Thus, we define term $\mathcal{L}_{dist}$ as

$$\begin{aligned}\mathcal{L}_{dist} &= \mathcal{L}_{same} - \rho \mathcal{L}_{diff} \\ &= \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X} (\mathbf{D}_{same} - \rho \mathbf{D}_{diff}) \mathbf{X}^\top \mathbf{P} \right), \end{aligned} \tag{16}$$

where the scalar ρ in (16) is used for balancing the two distance terms. If we minimize $\mathcal{L}_{dist}$, data with the same label will gather together and different clusters will be pulled away, which result in the learned features being discriminative enough.

D. Optimization

The intuition behind DICD is to learn both domain invariant and class discriminative feature representations, which will simultaneously draw source and target data closer and enhance the final classification performance in a latent subspace. Therefore, we can incorporate (5) and (16) into one object function as follows:

$$\begin{aligned} &\min_{\mathbf{P}} \ \mathcal{L}_{MMD} + \alpha \mathcal{L}_{dist} + \beta \left\| \mathbf{P} \right\|_F^2 \\ &\text{s.t. } \mathbf{P}^\top \mathbf{X} \mathbf{H} \mathbf{X}^\top \mathbf{P} = \mathbf{I}_d, \end{aligned} \tag{17}$$

where α and β are two tradeoff parameters. $\mathbf{I}_d$ is a identity matrix of dimension d, and $\mathbf{H}$ in the constraints is called centering matrix, which is 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 constraint of (17) is derived from the famous Principle Component Analysis (PCA) [38] method, which aims to maximize the embedded data variance. Thus, the constraint in (17) will preserve the data properties as much as possible after adaptation, and could keep the intrinsic information of both domains in the latent space.

Notably, if $\alpha = 0$, DICD reduces to JDA. However, in this paper, we believe the domain invariance and class discriminativeness of the learned representations are both of vital importance. The matrixes $\mathbf{W}, \mathbf{D}_{same}, \mathbf{D}_{diff}$ can be normalized respectively, which leads to the scales of $\mathcal{L}_{MMD}$ and $\mathcal{L}_{dist}$ be roughly in the same order of magnitude. Therefore, we empirically set $\alpha = 1$ for all the cases.

If we incorporate the specific forms of (5), (16) and $\alpha = 1$ into (18), We can rewrite (17) as:

$$\begin{aligned} &\min_{\mathbf{P}} \ \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{X} \left(\mathbf{W} + \mathbf{D}_{same} - \rho \mathbf{D}_{diff}\right) \mathbf{X}^\top \mathbf{P}\right) + \beta \left\| \mathbf{P} \right\|_F^2 \\ &\text{s.t. } \mathbf{P}^\top \mathbf{X} \mathbf{H} \mathbf{X}^\top \mathbf{P} = \mathbf{I}_d. \end{aligned} \tag{18}$$

Obviously, (18) is a constrained nonlinear optimization problem, and it is convenient to apply Lagrange techniques and derive the corresponding Lagrangian function for problem (18) as:

$$\begin{aligned} L(\mathbf{P}, \Theta) = &\ \mathbf{Tr}\left(\mathbf{P}^\top \left(\mathbf{X}\Omega\mathbf{X}^\top + \beta \mathbf{I}_m\right)\mathbf{P}\right) \\ &+ \mathbf{Tr}\left(\left(\mathbf{I}_d - \mathbf{P}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{P}\right)\Theta\right), \end{aligned} \tag{19}$$

where we denote $\Omega = \mathbf{W} + \mathbf{D}_{same} - \rho \mathbf{D}_{diff}$, and $\Theta = \mathrm{diag}(\theta_1, \theta_2, ..., \theta_d) \in \mathbb{R}^{d\times d}$ is a diagonal matrix with Lagrange Multipliers. By setting the gradient of $L(\mathbf{P}, \Theta)$ with respect to $\mathbf{P}$ to zero, we have

$$\left(\mathbf{X}\Omega\mathbf{X}^\top + \beta \mathbf{I}_m\right)\mathbf{P} = \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{P}\Theta, \tag{20}$$

Clearly, the problem expressed in (18) - (20) is a typical generalised eigen-decomposition problem, which can be effectively and efficiently solved by calculating the eigenvectors of (20) corresponding to the d-smallest eigenvalues. Finally, we summarise DICD in Algorithm 1.

E. Remarks

We have some remarks on the DICD approach:

- DICD could have different implementations by choosing whether ρ is equal to 0:

  (1) If $\rho = 0$, we denote our approach as DICD-S, which only constrains the distance between instances in the same class;

  (2) If $\rho \neq 0$, we call the approach as DICD, which jointly minimizes the intra-class variations and maximizes the inter-class variations explicitly across the projected source and target domains.
- **Kernelization:** Similar to [10], [12], DICD can solve nonlinear problems by kernelization. If we denote the transformation ψ as a kernel mapping $\psi : \boldsymbol{x} \to \psi(\boldsymbol{x})$, the kernel matrix of source and target data can be presented by the kernel tricks, i.e. $\mathbf{K} = \psi(\mathbf{X})^\top \psi(\mathbf{X}) \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$. By applying the empirical kernel map [39], we can formulate the nonlinear version of DICD as:

$$\begin{aligned} &\min_{\mathbf{P}} \ \mathbf{Tr}\left(\mathbf{P}^\top \mathbf{K} \left(\mathbf{W} + \mathbf{D}_{same} - \rho \mathbf{D}_{diff}\right) \mathbf{K}^\top \mathbf{P}\right) + \beta \left\| \mathbf{P} \right\|_F^2 \\ &\text{s.t. } \mathbf{P}^\top \mathbf{K}\mathbf{H}\mathbf{K}^\top \mathbf{P} = \mathbf{I}_d. \end{aligned} \tag{21}$$

  Utilizing the similar optimization scheme as (18), the adaptive matrix $\mathbf{P}$ of nonlinear case will be derived successfully.
- **Complexity Analysis:** Here, we would analyze the computational complexity of Algorithm **??** in detail. For Step 1 and Step 4, constructing matrix Ω would cost around $\mathcal{O}\left((NC + 3N)(n_s + n_t)^2\right)$; For Step 2, solving the generalised eigen-decomposition problem would take $\mathcal{O}(Ndm^2)$; For Step 3 and all other steps, it needs around $\mathcal{O}\left(Ndm(n_s + n_t)\right)$. In summary, the whole computational complexity of DICD is $\mathcal{O}\left((NC + 3N)(n_s + n_t)^2 + Ndm^2 + Ndm(n_s + n_t)\right)$. In practice, the iteration number N and the typical value of d are usually small, i.e. $N, d \ll \min(m, n_s+n_t)$, which is not a big issue for the practice operation time.

Algorithm 1 DICD Algorithm

Input:

Labeled source samples, $\{\mathbf{X}_S, \boldsymbol{y}_S\} = \{(\boldsymbol{x}_{Si}, y_{Si})\}_{i=1}^{n_s}$;
Unlabeled target samples, $\{\mathbf{X}_T\} = \{\boldsymbol{x}_{Tj}\}_{j=1}^{n_t}$;
Tradeoff parameters: $\alpha = 1$ ρ, β; Iterations: N;
Number of latent subspace dimension: d.

1: Construct matrix $\Omega = \mathbf{W}_0 + \mathbf{D}_{same} - \rho \mathbf{D}_{diff}$, where $\mathbf{D}_{same} = \mathrm{diag}\left(\mathbf{D}_{same}^{(S)}, \mathbf{0}_{n_t\times n_t}\right)$, $\mathbf{D}_{diff} = \mathrm{diag}\left(\mathbf{D}_{diff}^{(S)}, \mathbf{0}_{n_t\times n_t}\right)$.

while not converge **or** $t \leqslant N$ **do**

2: Derive the projected matrix $\mathbf{P}$ by computing the d-smallest eigenvectors of Eq. (20).

3: Let $[\mathbf{Z}_S, \mathbf{Z}_T] = \mathbf{P}^\top [\mathbf{X}_S, \mathbf{X}_T]$, and train a standard classifier f on $\{\mathbf{Z}_S, \boldsymbol{y}_S\}$ to predict target pseudo label $\hat{\boldsymbol{y}}_T$.

4: Update matrix $\Omega = \mathbf{W} + \mathbf{D}_{same} - \rho \mathbf{D}_{diff}$.

5: $t = t + 1$.

end while

Output:

Projected matrix $\mathbf{P}$, and the final classifier f.

Fig. 2. Image samples from CMU-PIE, MNIST, USPS, COIL20, Office and Caltech-256 datasets. Details are described in Section IV-A.

TABLE II
DESCRIPTION OF VISUAL CROSS-DOMAIN DATASETS USED IN THE EXPERIMENTS

Dataset	Type	#Samples	#Classes	#Features
CMU-PIE	Face	11554	68	1024
AMAZON (A)	Object	958	10	800/4096
CALTECH (C)	Object	1123	10	800/4096
DSLR (D)	Object	157	10	800/4096
WEBCAM (W)	Object	295	10	800/4096
COIL20	Object	1440	20	1024
MNIST	Digit	2000	10	256
USPS	Digit	1800	10	256

IV. EXPERIMENTS AND RESULTS

In this section, we compare our approach with two conventional machine learning methods, i.e. 1-nearest neighbor (1-NN) [17], Principal Component Analysis (PCA) [38], and eight state-of-the-art domain adaptation methods: Geodesic flow kernel (GFK) [9], Transfer Component Analysis (TCA) [12], Joint Domain Adaptation (JDA) [10], Transfer Joint Matching (TJM) [11], Transfer Subspace Learning (TSL) [40], Discriminative Transfer Subspace Learning (DTSL) [41], Cross-Domain Metric Learning (CDML) [33] and Robust Transfer Metric Learning (RTML) [14], on several real-world image cross-domain benchmark datasets. Datasets description and experimental implementation details will be introduced first.[1]

A. Datasets Description

CMU-PIE [42], Office [9], [10], [43], Caltech256 [10], [44], COIL20 [45], MNIST [10], [11], and USPS [10], [11] are widely used cross-domain face, object and digit datasets, respectively. The image samples are shown in Fig. 2 and the description is displayed in Table II. We will introduce them in detail as follows.

CMU-PIE[2] CMU Pose, Illumination, and Expression (PIE) dataset consists of more than 40,000 face images with resolution of 32×32 from 68 individuals. Based on the different pose factors, we choose five subsets: C05 (left pose), C07 (upward pose), C09 (downward pose), C27 (front pose) and C29 (right pose) to construct the cross-domain classification tasks. Since images in the different subsets may vary a lot and follow significant different distributions [10], [42], we can select two different subsets as the source and target domains to thoroughly evaluate our approach. In this way, we can construct 20 cross-domain tasks, e.g. "C05→C07," "C05→C09," …, "C29→C27" as source→target.

Office+Caltech-256[3] Office dataset contains more than 4000 images with 31 categories of common office objects, i.e. chairs, keyboards, monitor, etc. The dataset consists of three distinct object domains: Amazon (images from amazon.com), WEBCAM (images taken in "the wild" by a web camera [46]) and DSLR (images taken by a digital SLR camera). Caltech is another famous standard dataset for object recognition with 256 categories over 30,000 samples.

In our experiments, we choose to use the Office+Caltech256 dataset released in [9] and [47], which contains 10 shared category objects. We adopt two kinds of features: 800-dim SURF features [9] and 4096-dim DeCAF_6 [47] features. In the end, we have four domains: i.e. AMAZON (A), CALTECH (C), DSLR (D) and WEBCAM (W) to construct the cross-domain tasks. Specifically, by randomly choosing two domains as the source and target, we can obtain 12 cross-domain tasks for two kinds of features, e.g. "A → C," "A → D," ..., "W → C" and "W → D."

COIL20[4] COIL20 dataset consists of 20 objects with 1440 grayscale images. For each object, there are 72 images of size 32×32 taken at pose intervals of 5 degrees rotating through 360 degrees. As [10], we split the dataset into 2 subsets: COIL1 and COIL2. To be specific, COIL1 includes all the images at the directions of $[0°, 85°] \cup [180°, 265°]$, and COIL2 contains images of $[90°, 175°] \cup [270°, 355°]$. Since COIL1 and COIL2 consist of the same objects with diverse shooting degrees, they follow different but related distributions. We could choose one as the source domain and the other as the target domain to construct two cross-domain tasks, e.g. "COIL1 → COIL2" and "COIL2 → COIL1."

MNIST+USPS[5] MNIST and USPS are two famous digital datasets sharing ten handwritten digits from 0 to 9. MNIST dataset contains 60,000 training data and 10,000 test samples, and USPS dataset has 7,291 training images and 2007 test samples. From Fig. 2, we can see the distributions of MNIST and USPS are very different. Follow the technique of [10], [11], to speed up experiments, we randomly choose 1,800 images from USPS and 2,000 images from MNIST, and rescale all of them to size 16×16 to form our cross-domain tasks, e.g. "MNIST → USPS" and "USPS → MNIST."

B. Experiments Implementation Details

To extensively evaluate our approach, we compare DICD with 10 machine learning methods: 1-NN [17], PCA [38], GFK [9], TCA [12], JDA [10], TJM [11], TSL [40], DTSL [41], CDML [33] and RTML [14], which are base classifier, unsupervised dimension reduction method, domain-invariant feature extraction methods and adaptive metric learning methods, respectively.

Under the unsupervised domain adaptation experiment settings [10], [12], we can only access labeled source data and unlabeled target samples, and the optimal parameters

[1]The codes and datasets will be uploaded online once the paper is published.
[2]http://www.ri.cmu.edu/research_project_detail.html?project_id=418&menu_id=261
[3]http://www-scf.usc.edu/~boqinggo/domainadaptation.html
[4]http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[5]http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

TABLE III
AVERAGE CLASSIFICATION ACCURACY (%) OF CMU-PIE FROM SOURCE TO TARGET

Tasks\Methods	NN	PCA	GFK	TCA	JDA	TSL	DTSL	CDML	RTML	DICD-S	DICD
C05→C07	26.09	24.80	26.15	40.76	58.81	44.08	65.87	53.22	60.12	64.76	**72.99**
C05→C09	26.59	25.18	27.27	41.79	54.23	47.49	64.09	53.12	55.21	61.52	**72.00**
C05→C27	30.67	29.26	31.15	59.63	84.50	62.78	82.03	80.12	85.19	90.81	**92.22**
C05→C29	16.67	16.30	17.59	29.35	49.75	36.15	54.90	48.23	52.98	55.02	**66.85**
C07→C05	24.49	24.22	25.24	41.81	57.62	46.28	45.04	52.39	58.13	64.44	**69.93**
C07→C09	46.63	45.53	47.37	51.47	62.93	57.60	53.49	54.23	63.92	64.40	**65.87**
C07→C27	54.07	53.35	54.25	64.73	75.82	71.43	71.43	68.36	76.16	83.99	**85.25**
C07→C29	26.53	25.43	27.08	33.70	39.89	35.66	47.97	37.34	40.38	45.04	**48.71**
C09→C05	21.37	20.95	21.82	34.69	50.96	36.94	52.49	43.54	53.12	53.54	**69.36**
C09→C07	41.01	40.45	43.16	47.70	57.95	47.02	55.56	54.87	58.67	59.98	**65.44**
C09→C27	46.53	46.14	46.41	56.23	68.45	59.45	77.50	62.76	69.81	77.35	**83.39**
C09→C29	26.23	25.31	26.78	33.15	39.95	36.34	54.11	38.21	42.13	50.25	**61.40**
C27→C05	32.95	31.96	34.24	55.64	80.58	63.66	81.54	75.12	81.12	89.47	**93.13**
C27→C07	62.68	60.96	62.92	67.83	82.63	72.68	85.39	80.53	83.92	88.21	**90.12**
C27→C09	73.22	72.18	73.35	75.86	87.25	83.52	82.23	83.72	**89.51**	88.66	88.97
C27→C29	37.19	35.11	37.38	40.26	54.66	44.79	72.61	52.78	56.26	62.99	**75.61**
C29→C05	18.49	18.85	20.35	26.98	46.46	33.28	52.19	27.34	29.11	54.98	**62.88**
C29→C07	24.19	23.39	24.62	29.90	42.05	34.13	49.41	30.82	33.28	49.29	**57.03**
C29→C09	28.31	27.21	28.49	29.90	53.31	36.58	58.45	36.34	39.85	56.80	**65.87**
C29→C29	31.24	30.34	31.33	33.64	57.01	38.75	64.31	40.61	47.13	63.95	**74.77**
average	34.76	33.85	35.35	44.75	60.24	49.43	63.53	53.68	58.80	66.27	**73.09**

for the target classifier can't be selected by cross validation procedure. Therefore, as [14], [27], and [41], we evaluate all the baselines utilizing the strategy that grid-searches their hyper-paremeter space, and the best results on our cross-domain tasks are reported. For PCA, GFK, TCA, JDA, TJM, TSL, DTSL, we decide their optimal dimension of common subspace by searching $d = \{10, 20, ..., 100\}$. For CDML and RTML, we train the transfer metric according to [14] and [33]. Then, labeled source samples are adopted to learn the target classifier in the latent subspace or under the transfer metric, and unlabeled target data are exploited to calculate the final accuracy.

In this paper, we use 1-nearest neighbor (1-NN) classifier as the final classifier for DICD. Also, for fair comparison, we choose 1-NN as the base classifier for all the other baselines in the experiments since we don't need to tune the hyper parameters.[6]

We claim that domain-invariance and discriminativeness of the learned representations are both critical for domain adaptation problem. Thus, we treat $\alpha = 1$ in DICD as a fixed parameter for all the experiments. Then there are two tunable parameters: tradeoff parameter ρ to balance the effects of intra-class compactness and inter-class dispersion, and regularization parameter β. Similar to TCA and JDA, as recommended in the literature [10], we set $\beta = 1$ for datasets Office+Caltech-256 (both SURF and DeCAF$_6$ features), and $\beta = 0.1$ for other datasets in DICD. For ρ, we empirically set $\rho = 1$ for the face datasets, i.e. CMU-PIE, and $\rho = 0.1$ for other datasets. In the subsequent Section IV-D, we will conduct detailed parameter sensitivity experiments to verify that DICD can achieve comparable results at a wide range of parameter values. Here, we fix the number of iterations as $N = 10$.

[6]For partial experimental results, we also quote from [10], [11], [14], and [41] if the experiments settings are identical with ours.

C. Experimental Results

In this section, we comprehensively evaluate DICD and all the baselines on several visual cross-domain tasks (including face, object and digit datasets) in terms of the classification accuracy to demonstrate the effectiveness of the proposed algorithm.

1) Results of CMU-PIE: We first experiment on CMU-PIE datasets, and the results are illustrated in Table III, in which the first and second best results are marked in bold and underlined, repectively. From Table III, DICD achieves much better performance than all the other baselines. To be specific, the average classification accuracy of DICD is **73.09**%, and DICD gains a significant performance improvement of **9.56**% compared to the best baseline DTSL. It's worth noting that, DICD obtains the best result in almost all the tasks (19 out of 20) with a large margin improvement. Since the average result of 1-NN is only 34.76%, this manifests the difficulty of these adaptation tasks, and also presents the effectiveness and robustness of DICD.

Secondly, we notice that the performance of domain adaptation methods is better than the standard machine learning methods. This phenomenon shows the importance of mitigating the domain shift when the training and test data are drawn from different distributions. JDA is better than GFK, TCA and TSL, because JDA jointly reduces the marginal and conditional distributions between source and target. JDA can deal with more tough situations. For transfer metric learning methods, RTML can consistently outperform CDML, which aims to align the marginal distribution across two domains by minimizing KL-divergence. But RTML reduces the discrepancy between source and target incorporating domain-wise and class-wise adaptation terms. This strategy leads to RTML being more adaptive real-world tasks.

However, we believe that the domain invariance and class discriminativeness of the learned representations are equally

TABLE IV

AVERAGE CLASSIFICATION ACCURACY (%) OF OFFICE+CALTECH-256 (SURF FEATURES), MNIST+USPS AND COIL20 DATASETS FROM SOURCE TO TARGET, WHERE A = AMAZON, C = CALTECH, D = DSLR AND W = WEBCAM

Tasks\Methods	NN	PCA	GFK	TCA	JDA	TJM	DTSL	CDML	RTML	DICD-S	DICD
C→A (SURF)	23.70	36.95	41.02	38.20	44.78	46.76	**51.25**	47.82	49.26	47.18	47.29
C→W (SURF)	25.76	32.54	40.68	38.64	41.69	38.98	38.64	36.91	44.72	**48.47**	46.44
C→D (SURF)	25.48	38.22	38.85	41.40	45.22	44.59	47.13	43.93	47.56	**49.68**	**49.68**
A→C (SURF)	26.00	34.73	40.25	37.76	39.36	39.45	43.37	41.72	**43.68**	42.12	42.39
A→W (SURF)	29.83	35.59	38.98	37.63	37.97	42.03	36.61	38.25	44.32	44.41	**45.08**
A→D (SURF)	25.48	27.39	36.31	33.12	39.49	**45.22**	38.85	35.92	43.86	39.49	38.85
W→C (SURF)	19.86	26.36	30.72	29.30	31.17	30.19	29.83	31.14	**34.83**	33.57	33.57
W→A (SURF)	22.96	31.00	29.75	30.06	32.78	29.96	34.13	32.26	**35.28**	34.13	34.13
W→D (SURF)	59.24	77.07	80.89	87.26	89.17	89.17	82.80	84.84	**91.02**	90.45	89.81
D→C (SURF)	26.27	29.65	30.28	31.70	31.52	31.43	30.11	32.63	34.58	33.93	**34.64**
D→A (SURF)	28.50	32.05	32.05	32.15	33.09	32.78	32.05	29.87	33.26	**34.45**	**34.45**
D→W (SURF)	63.39	75.93	75.59	86.10	89.49	85.42	72.20	82.34	89.68	90.85	**91.19**
USPS→MNIST	44.70	44.95	46.45	51.05	59.65	52.25	55.50	52.25	61.82	61.05	**65.20**
MNIST→USPS	65.94	66.22	67.22	56.28	67.28	63.28	52.33	63.28	69.52	**77.83**	**77.83**
COIL1→COIL2	83.61	84.72	72.50	88.47	89.31	91.53	88.61	88.93	91.23	94.31	**95.69**
COIL2→COIL1	82.78	84.03	74.17	85.83	88.47	91.81	89.17	87.32	90.22	91.81	**93.33**
average	40.84	47.34	48.48	50.31	53.78	53.43	51.41	51.84	56.55	57.11	**57.47**

vital for domain adaptation. The discriminability will help transfer the source knowledge (especially the discriminative information) to the target more successfully. Thus, DICD can generalize well on the unlabeled target samples.

Lastly, DICD performs better than DICD-S, which only requires to minimize the intra-class variance for source and target. However, only clustering the samples sharing the same label is far from effective for classification. The inter-class difference also should be maximized to construct the discriminative features. While, the classification accuracy of DICD-S is still higher than other baselines, which further proves the importance of learning class discriminative features.

The detailed analysis about DICD and baselines will be conducted in Section IV-D, and we will visualize the effectiveness of our approach through data embedding similarity matrix and MMD distance calculation.

2) Results of Office+Caltech-256, MNIST+USPS and COIL20: The average classification accuracy of DICD and other baselines on the object (COIL20, Office+Caltech-256, SURF and DECAF$_6$ features) and digit (MNIST+USPS) datasets are listed in Table IV and Table V. DICD generally outperforms other baselines on most tasks. Here, we select TJM as a new baseline instead of TSL, since TJM can obtain better results than TSL in these datasets.

We notice that the adaptation difficulty varies a lot on these datasets, and the results of RTML and JDA are superior to other baselines. The main reasons are two-folds: 1) they all focus on matching both marginal and conditional distributions across two domains, and mitigate the domain shift more effectively; 2) RTML utilizes marginalized denoising scheme to reconstruct the data, which makes RTML more robust to different cross-domain tasks. DTSL performs pretty well on face dataset, but poorly on object and digit datasets. We conjecture the reasons are that the reconstruction strategy in DTSL is not very effective when the discrepancy between source and target is tremendous, and the tradeoff parameters in DTSL for different tasks should be selected delicately.

However, DICD is designed to not only align the source and target domains as much as possible, but also explore the discriminative information of both domains deeply. DICD can effectively transfer the intrinsic source domain knowledge to target as well as boost the classification performance with a large margin.

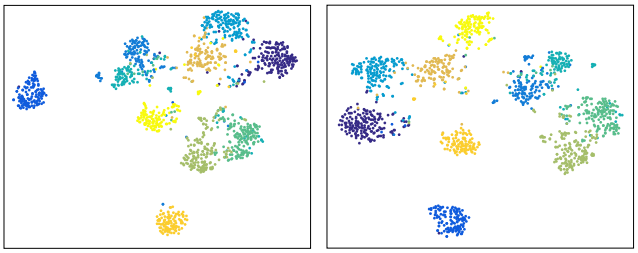


Fig. 3. Feature visualization: data embedding by t-SNE of (a) JDA features and (b) DICD features of the projected source and target data. Samples in different colors present they are in different categories.

We also observe that results of methods on DECAF$_6$ features are much better than on SURF features, which manifests that deep discriminative representations can bridge the domain discrepancy to some extent. Based on this, DICD still outperforms others, and the results demonstrate the effectiveness of DICD.

To present the domain invariance and class discriminativeness of the DICD learned features more clearly, we display the t-SNE [48] visualization plots in Fig. (3) for task C→A (DeCAF$_6$ features) of the projected representations derived by JDA and DICD, respectively. Comparing JDA and DICD features, we have the following observations: 1) with JDA features, the samples in both domains are not discriminative very well, especially there are several classes clustering together, which will degrade the performance dramatically; 2) with DICD features, both domains can be aligned very well, and the points are more discriminative, which implies the source classifier will predict the target data more correctly. Thus, DICD have more advantages for effective domain adaptation.

In real-world applications, we often confront with multi-domain adaptation problems. To fully evaluate

TABLE V

AVERAGE CLASSIFICATION ACCURACY (%) OF OFFICE+CALTECH-256 ($DECAF_6$ FEATURES) DATASETS FROM SOURCE TO TARGET, WHERE A = AMAZON, C = CALTECH, D = DSLR AND W = WEBCAM

Tasks\Methods	NN	PCA	GFK	TCA	JDA	TJM	DTSL	CDML	RTML	DICD-S	DICD
C→A ($DeCAF_6$)	85.70	88.10	87.79	89.46	89.77	89.46	**91.54**	86.24	90.62	91.13	91.02
C→W ($DeCAF_6$)	66.10	74.24	70.17	78.64	83.73	78.98	76.61	77.82	85.38	90.51	**92.20**
C→D ($DeCAF_6$)	74.52	83.44	88.54	81.53	86.62	85.35	87.90	83.74	89.32	**93.63**	**93.63**
A→C ($DeCAF_6$)	70.35	73.02	75.78	79.61	82.28	79.07	85.75	79.54	**86.43**	85.93	86.02
A→W ($DeCAF_6$)	57.29	57.63	76.95	73.22	78.64	76.95	73.56	76.27	80.26	80.68	**81.36**
A→D ($DeCAF_6$)	64.97	70.06	84.08	84.71	80.25	**85.35**	82.17	81.35	84.36	83.44	83.44
W→C ($DeCAF_6$)	60.37	63.58	75.07	78.09	83.53	76.49	72.75	77.64	83.13	**83.97**	**83.97**
W→A ($DeCAF_6$)	62.53	70.15	82.88	83.30	90.19	86.74	75.47	86.29	**91.37**	90.19	89.67
W→D ($DeCAF_6$)	98.73	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	98.42	**100.00**	**100.00**	**100.00**
D→C ($DeCAF_6$)	52.09	57.88	73.11	79.70	85.13	78.63	75.24	78.56	85.72	**86.11**	**86.11**
D→A ($DeCAF_6$)	62.73	68.16	85.18	88.52	91.44	89.77	84.97	89.47	91.86	92.07	**92.17**
D→W ($DeCAF_6$)	89.15	88.14	90.85	98.98	98.98	97.97	**99.32**	96.38	98.98	98.98	98.98
average	70.38	74.53	82.53	84.65	87.55	85.40	83.77	84.31	88.95	89.72	**89.88**

TABLE VI

AVERAGE CLASSIFICATION ACCURACY (%) OF OFFICE+CALTECH-256 ($DECAF_6$ FEATURES) DATASETS WITH MULTIPLE SUB-DOMAINS, WHERE A = AMAZON, C = CALTECH, D = DSLR AND W = WEBCAM

Tasks\Methods	NN	TCA	JDA	TJM	DTSL	RTML	DICD
A+C+D→W ($DeCAF_6$)	91.53	94.92	94.92	95.93	90.85	96.61	**98.64**
A+C+W→D ($DeCAF_6$)	98.09	97.45	97.45	**100.00**	98.73	99.36	**100.00**
C+D+W→A ($DeCAF_6$)	85.39	89.98	90.19	89.98	91.02	**92.38**	91.23
A+D+W→C ($DeCAF_6$)	73.73	80.94	83.17	80.50	**86.02**	83.70	85.75
D+W→A+C ($DeCAF_6$)	62.09	81.98	86.26	81.36	75.01	87.07	**87.99**
C+W→A+D ($DeCAF_6$)	87.09	91.21	91.93	91.12	**92.56**	92.04	**92.56**
C+D→A+W ($DeCAF_6$)	87.63	90.90	91.94	91.46	92.18	92.58	**93.06**
A+W→C+D ($DeCAF_6$)	76.88	82.58	84.84	82.19	86.80	85.16	**87.19**
A+D→C+W ($DeCAF_6$)	74.82	84.13	85.83	83.00	87.59	86.46	**88.01**
A+C→D+W ($DeCAF_6$)	67.92	78.76	81.64	82.30	81.86	86.06	**87.39**
average	80.52	87.29	88.82	87.78	88.26	90.14	**91.18**

TABLE VII

AVERAGES AND STANDARD ERRORS OF CLASSIFICATION ACCURACY (%) OF OFFICE-31 (31 CLASSES) DATASET FROM SOURCE TO TARGET, WHERE A = AMAZON, D = DSLR, W = WEBCAM. BEST RESULTS UP TO STATISTICAL SIGNIFICANCE ARE HIGHLIGHTED IN **BOLD**

Tasks\Methods	NN	PCA	TCA	JDA	TJM	DTSL	RTML	DICD
A→D	35.51±2.73	45.46±3.63	47.31±2.78	49.01±2.55	45.60±2.69	43.13±2.81	48.92±2.55	**54.02**±1.49
D→W	37.02±1.70	57.51±3.24	63.40±1.87	68.87±1.53	64.74±1.90	58.09±2.70	69.32±1.53	**74.43**±2.42
W→D	47.70±2.31	59.65±4.45	61.38±2.83	65.36±1.91	61.53±1.54	56.99±3.59	71.65±2.20	**74.10**±2.41

the generalization ability of DICD, we sample different sub-domains from Office+Caltech-256 ($DECAF_6$ features) datasets to construct 10 cross-domain data pairs (see Table VI). From Table VI, we can see that the source or target domains in each task contain multiple sub-domains data, and the most related and promising methods are compared. From the results, DICD achieves an average classification accuracy of 91.18%. It can be clearly observed that DICD is superior to other state-of-the-art domain adaptation baselines, and performs the best in 8 out of 10 tasks. Hence, in mix-domain scenarios, the effectiveness of DICD can be successfully verified.

D. Analytical Experiments

In this section, we will conduct several analytical experiments to verify the effectiveness of DICD.

1) Significance Test (t-Test) on Office-31 Dataset: In this part, we will verify the capability of DICD to deal with more complex domain adaptation scenarios. As introduced in Section IV-A, Office-31[7] dataset contains the images of 31 categories in the office environment collected from 3 distinct domains: Amazon(A), DSLR(R) and Webcam(W), and Office-31 is a more complex dataset [26], [50]. Following the classic protocol in [26] and [49], for the source domain, we randomly down-sample 20 labeled samples per class for Amazon and 8 for DSLR or Webcam. Further, for the target domain, we randomly select 3 target data as labeled samples, and the rest are used for unlabeled testing data. Similar to [26], [49], and [50], we construct three cross-domain tasks: "A → D," "D → W" and "W → D," and we use 800-bin SURF features of all the images in the experiments. All the experiments are repeated 10 times, and averages and standard errors of classification accuracy are showed in Table VII.

From Table VII, the best results up to statistical significance are highlighted in bold, and we can observe that DICD

[7]To distinguish with the Office+Caltech256 dataset [9], [47] used in Section IV-A, here we denote the Office dataset with 31 classes as Office-31 following [49], [50]. Since Office-31 has more categories, which results in the increasing of difficulty to classify them well.

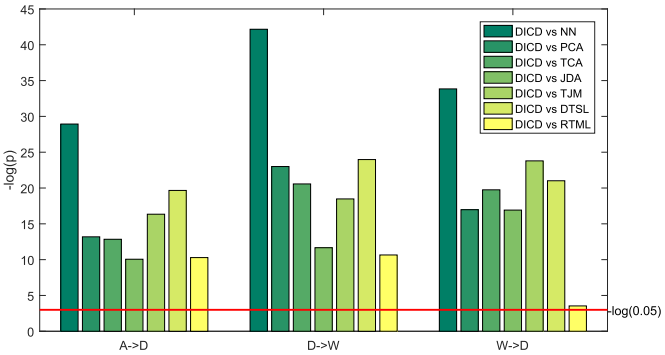


Fig. 4. p-value of the significance test (t-test) for results of DICD and other baselines on Office-31 dataset. To illustrate the statistical significance clearer, we have shown the $-log(p)$ with respect to each task and the base significance level of 0.05 ($-log(0.05)$) in red line. The larger value of $-log(p)$ means the more significance of DICD compared with other baselines.

TABLE VIII

AVERAGE CLASSIFICATION ACCURACY (%) COMPARISON OF DIFFERENT VARIANTS OF DICD

Tasks\Methods	JDA	DICD-S	DICD	DICD$_{no}$	DICD$_{whole}$
C05→C07	58.81	64.76	72.99	64.52	65.25
C07→C27	75.82	83.99	85.25	75.88	83.69
C→W (SURF)	41.69	48.47	46.44	45.42	45.42
C→W (DeCAF$_6$)	83.73	90.51	92.20	87.80	90.51
MNIST→USPS	67.28	71.22	77.83	73.11	67.56
COIL1→COIL2	89.31	94.31	95.69	94.03	91.67

performs much better than other baselines. To further prove the advantage of DICD to other methods for this more challenging dataset, for all the results, we conduct a significance test (t-test) which is illustrated in Fig. 4. Here, a significance level of 0.05 is used, and if the p-value is less than 0.05, the differences of results between DICD and other baselines are statistically significant. In Fig. 4, we have shown the $-log(p)$ of each p-value for clearer explanation. We can also observe that all the $-log(p)$ of the performance comparison between DICD and other methods for all the tasks are larger than $-log(0.05)$, which means DICD is significantly superior to other baselines at this complex dataset.

2) Variants of DICD: To understand more deeply about DICD, we propose several variants of DICD: 1) DICD$_{no}$ by removing the weights which aim to balance the effects of different categories, i.e. $\frac{n_s}{n_s^{(k)}} = \frac{n_t}{n_t^{(k)}} = 1$ for any class k in (6) and (8); 2) DICD$_{whole}$ by viewing source and target data as a whole part when we construct the distance matrixes $\mathbf{D}_{same}$ and $\mathbf{D}_{diff}$. To be specific, in the original DICD, we calculate intra-class and inter-class distance loss terms separately with respect to the source and target domains. DICD$_{whole}$ is to combine samples in both domains together, and treats target pseudo labels as their true labels. Then for any class k, there are $n_s^{(k)} + n_t^{(k)}$ source and target data blending as a whole group. $\mathbf{D}_{same}$ and $\mathbf{D}_{diff}$ can be correspondingly computed using the blended both domains data.

All these variant approaches are evaluated on different cases following the same experiment settings, and the results are listed in Table VIII. From Table VIII, we can notice that DICD also achieves the best result in these variant methods, which means the balance weights and the separate calculation of intra-class and inter-class distance loss are meaningful and effective for DICD. We also observe that all the variant methods of DICD perform much better than JDA. This emphasizes the importance to derive class discriminative features for domain adaptation problems.

3) Similarity Analysis: JDA, DICD-S and DICD are run utilizing their optimal parameters for task C05→C07 in CMU-PIE datasets to quantitatively verify the effectiveness of DICD.

We first construct the similarity matrix of JDA, DICD-S and DICD features. Each entry in the similarity matrix measures the similarity of the associated data pair from both domains. Here, we compute the inner product of all the data pairs to present the similarity matrix, since all the learned features have been normalized. To visualize the similarity matrix clearly, we apply a threshold of 0.5 to binarize the matrix, and the similarity plots are shown in Fig. 5. The top-left and bottom-right small squares illustrate the within-domain similarity of the source and target domains, respectively. Correspondingly, the top-right and bottom-left submatrices present the between-domain similarity between source and target.

Firstly, Comparing (a) and (c), it is clear that a large number of irrelevant samples are too similar to distinguish them correctly for JDA features. However, DICD features could compact points in the same category, which leads to the within-class similarity being high, and disperse points sharing different classes as much as possible, which results in the between-class similarity being low enough. Meanwhile, DICD features could align the same class samples as well as decouple the between-class data correlation in the source and target domains pretty well. This strategy improves the generalization ability of DICD features across domains a lot. Secondly, Comparing (a) and (b), we can observe that DICD-S features are more discriminative than JDA features only by clustering the same class data, which means the class discriminative information is of vital importance for extracting an ideal representations. Lastly, the difference between (b) and (c) demonstrates the effectiveness of maximizing the inter-class dispersion of both domains for domain adaptation problems.

4) MMD Distance: We then compute the MMD distance [10], [14] and classification accuracy of the learned TCA, JDA, DICD-S and DICD features with respect to iterations for task C05→C07, which are showed in Fig. 6. It is worth noting that the smaller distribution distance (e.g. MMD distance) of the derived representations will lead to a better generalization performance for the target data. From Fig. 6(a), we can clearly see that JDA features can explicitly reduce the MMD distance by matching both marginal and conditional distributions across two domains compared to TCA features. However, the MMD distance of DICD features is the smallest. The reason is that DICD features are not only domain invariant but also class discriminative, and these two properties will help DICD features align the source and target domains much better.

Fig. 6(b) illustrates the corresponding classification accuracy of the compared methods. DICD achieves much better results than TCA, JDA and DICD-S from the beginning

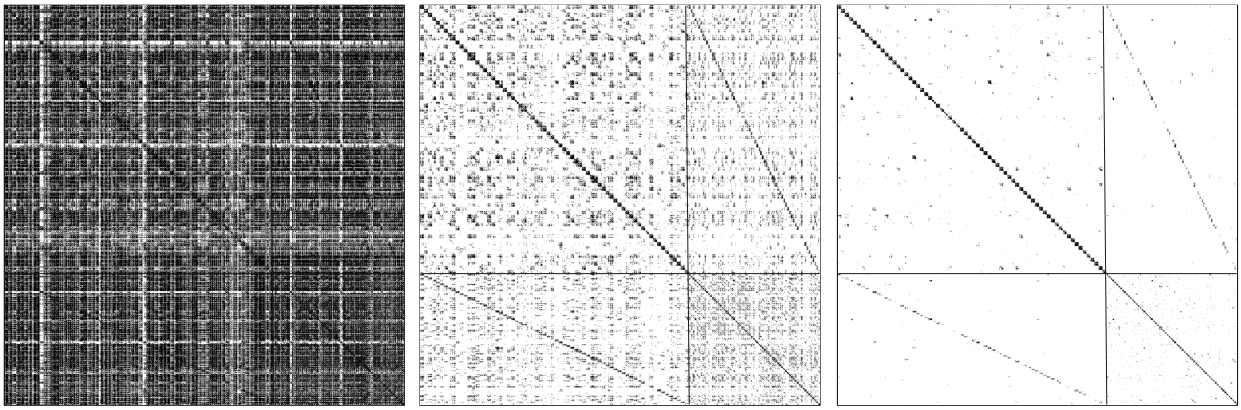


Fig. 5. Domain similarity visualization for task C05→C07 of JDA features, DICD-S features and DICD features.

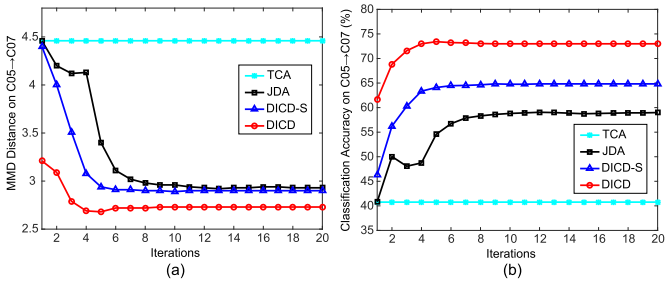


Fig. 6. MMD distance and classification accuracy curves for task C05→C07 of TCA, JDA, DICD-S and DICD.

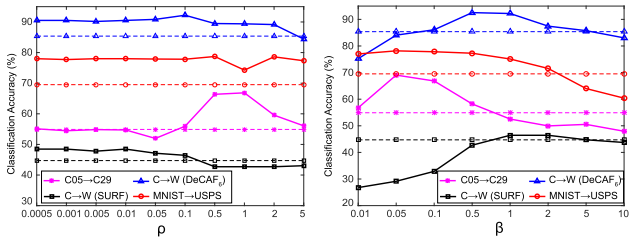


Fig. 7. Parameter sensitivity study on task C05→C29, C→W (SURF), C→W (DeCAF$_6$) and MNIST→USPS. The dashed lines show the corresponding best baseline results.

iteration, and gets converged after several iterations, which implies DICD features will reach a stable condition quickly.

If we observe Fig. 5 and 6 together, we will conclude that to learn domain invariant features and class discriminative features are complementary to each other. The class discriminative information of both domains should be leveraged to learn ideal cross-domain feature representations.

5) Parameter Sensitivity: As mentioned in Section III-D, we empirically set $\alpha = 1$ for all the experiments. Then there are only two tunable parameters: ρ and β for DICD. We have conducted extensive parameter sensitivity analysis on all face, object and digit datasets, and results of task C05→C29, C→W (SURF), C→W (DeCAF$_6$) and MNIST→USPS are reported in Fig. 7. Meanwhile, to illustrate the effectiveness of DICD, best baseline results are shown in Fig. 7 as the dashed lines.

From Fig. 7, we can see that DICD could achieve much better performance than baselines under a wide range of parameter values. We first run DICD as ρ varies from 0.0005 to 5, with β fixed. It can be observed that ρ with small value will help DICD improve the classification accuracy, but when ρ becomes too large, it will dominate the $\mathcal{L}_{dist}$ term in (16), which will weaken the effects of clustering term and degrade the classification performance.

We then evaluate DICD by fixing ρ, and varying the regularization parameter β from 0.01 to 10. In principle, smaller values of β will lead to the ill-defined optimization problem, and larger values of β could avoid DICD learning perfect domain transfer features. The bell-shaped curves in Fig. 7 also verify our analysis. A reasonable β will make DICD better than other baselines generally. Note that, in practice DICD is less sensitive to ρ than β, and this guides us to determine $\rho \in [0.001, 1]$ first. Even if $\rho = 0$, DICD will reduce to DICD-S, which also achieves much better results than other baselines.

V. Conclusion

In this paper, we have introduced a novel feature extraction approach to learn both domain invariant and class discriminative (DICD) representations for visual domain adaptation problems. In particular, DICD is designed to simultaneously reduce the distance of marginal and conditional distributions across domains, as well as jointly minimize the within-class variation and maximize the between-class scatter for the source and target domains. All these goals can be incorporated into one optimization problem, and the global optimal solution could be obtained. Analytical experiments have been conducted and proven that the domain invariance and class discriminativeness of learned features are complementary to each other, and are both of vital importance to learn a robust and perfect cross-domain representations. Extensive experiments on visual cross-domain datasets (CMU-PIE, COIL20, USPS, MNIST, Office and Caltech256) have convincingly demonstrate that DICD is superior to several state-of-the-are domain adaptation methods.

References

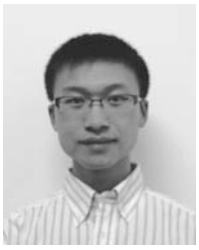
[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. AAAI*, vol. 8, 2008, pp. 677–682.

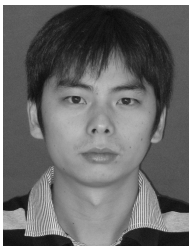
[3] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 222–230.
[4] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.
[5] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, "Distant domain transfer learning," in *Proc. AAAI*, 2017, pp. 1–7.
[6] W.-C. Chang, Y. Wu, H. Liu, and Y. Yang, "Cross-domain kernel induction for transfer learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1763–1769.
[7] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2456–2464.
[8] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 769–776.
[9] 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.
[10] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.
[11] 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.
[12] 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.
[13] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2980–2989, Oct. 2011.
[14] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.
[15] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
[16] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2007, pp. 1–8.
[17] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Trans. Comput.*, vol. TC-24, no. 7, pp. 750–753, Jul. 1975.
[18] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.
[19] C.-W. Hsu, C.-C. Chang, and C.-J. Li, "A practical guide to support vector classification," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2003.
[20] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[21] J. Huang *et al.*, "olkopf, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1–8.
[22] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, May 2007.
[23] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1682–1695, Jul. 2017.
[24] R. Xia *et al.*, "Instance selection and instance weighting for cross-domain sentiment classification via PU learning," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2176–2182.
[25] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2481–2488.
[26] L. Zhang, W. Zuo, and D. Zhang, "LSDT: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1177–1191, Mar. 2016.
[27] 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.
[28] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.
[29] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 188–197.
[30] P. Cui, S. Liu, and W. Zhu, "General knowledge embedded image representation," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 198–207, Jan. 2018.
[31] Z. Shen, P. Cui, K. Kuang, and B. Li. (2017). "On image classification: Correlation v.s. Causality." [Online]. Available: https://arxiv.org/abs/1708.06656
[32] 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.
[33] H. Wang, W. Wang, C. Zhang, and F. Xu, "Cross-domain metric learning based on information theory," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2099–2105.
[34] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 325–333.
[35] D. Zhang, Z.-H. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2007, pp. 629–634.
[36] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.
[37] J. P. Hwang, S. Park, and E. Kim, "A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8580–8585, 2011.
[38] I. T. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.
[39] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
[40] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.
[41] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 850–863, Feb. 2016.
[42] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 46–51.
[43] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 74–93, 2014.
[44] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694
[45] S. A. Nene *et al.*, "Columbia object image library (coil-20)," Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-96, 1996.
[46] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. (2013). "Efficient learning of domain-invariant image representations." [Online]. Available: https://arxiv.org/abs/1301.3224
[47] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 32, Jun. 2014, pp. 647–655.
[48] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[49] 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.
[50] M. Long, Y. Cao, J. Wang, and M. I. Jordan. (2015). "Learning transferable features with deep adaptation networks." [Online]. Available: https://arxiv.org/abs/1502.02791v2

Shuang Li received the B.S. degree with the Department of Automation, Northeastern University, in 2012. He is currently pursuing the Ph.D. degree with the Department of Automation, Institute of System Integration, Tsinghua University, China. He was a Visiting Research Scholar with the Department of Computer Science, Cornell University, from 2015 to 2016. His main research interests include machine learning and pattern recognition, especially in transfer learning and domain adaptation.

Shiji Song received the Ph.D. degree from the Department of Mathematics, Harbin Institute of Technology, in 1996. He is currently a Professor with the Department of Automation, Tsinghua University. His research interests include system modeling, control and optimization, computational intelligence, and pattern recognition.



Gao Huang received the B.S. degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2009, and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, in 2015. He is currently a Post-Doctoral Researcher with the Department of Computer Science, Cornell University. His current research interests include machine learning, statistical pattern recognition, and deep learning.



Zhengming Ding (S'14) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Northeastern University, USA. His research interests include machine learning and computer vision. He received the National Institute of Justice Fellowship.

Cheng Wu received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China.

Since 1967, he has been with Tsinghua University, where he is currently a Professor with the Department of Automation. His current research interests include system integration, modeling, scheduling, and optimization of complex industrial systems.

Mr. Wu is a member of the Chinese Academy of Engineering.