Aggregating Randomized Clustering-Promoting Invariant Projections for Domain Adaptation

Jian Liang, Ran He, *Senior Member, IEEE,* Zhenan Sun, *Member, IEEE,* and Tieniu Tan, *Fellow, IEEE*

Abstract—Unsupervised domain adaptation aims to leverage the labeled source data to learn with the unlabeled target data. Previous tranducstive methods tackle it by iteratively seeking a low-dimensional projection to extract the invariant features and obtaining the pseudo target labels via building a classifier on source data. However, they merely concentrate on minimizing the cross-domain distribution divergence, while ignoring the intra-domain structure especially for the target domain. Even after projection, possible risk factors like imbalanced data distribution may still hinder the performance of target label inference. In this paper, we propose a simple yet effective domain-invariant projection ensemble approach to tackle these two issues together. Specifically, we seek the optimal projection via a novel relaxed domain-irrelevant clustering-promoting term that jointly bridges the cross-domain semantic gap and increases the intra-class compactness in both domains. To further enhance the target label inference, we first develop a 'sampling-and-fusion' framework, under which multiple projections are independently learned based on various randomized coupled domain subsets. Subsequently, aggregating models such as majority voting are utilized to leverage multiple projections and classify unlabeled target data. Extensive experimental results on six visual benchmarks including object, face, and digit images, demonstrate that the proposed methods gain remarkable margins over state-of-the-art unsupervised domain adaptation methods.

Index Terms—Unsupervised domain adaptation, domain-invaraint projection, class-clustering, sampling-and-fusion.

1 Introduction

Traditional machine learning methods often assume that the training and testing data lie in the same feature space and have the same distribution [1]. However, this assumption does not always hold in massive real-world scenarios. For example, it is really challenging to recognize the adult faces while exploiting a set of labeled face images captured from their childhood. When this assumption is not verified, domain shift or covariate shift (i.e., the distributions of training and testing data are not identical when the conditional distributions are the same) largely affects the performance at test time. To address this issue, acquisition of annotated data (e.g., adult faces) is critical, however, data labeling is expensive and time-consuming. Hence one can resort to another strategy, transfer learning, which tries to explore the heterogeneous knowledge hidden in target data. Recently, considerable research efforts have been devoted to this topic, and impressive progress has been made in a wide range of applications, e.g., computer vision [2], [3], [4], [5], natural language processing [6], [7], [8]. Typically, unsupervised domain adaptation (DA) that aims to transfer the same task from supervised source domains to unsupervised target domains, has drawn increasing attention in computer vision literature [4], [5], [9], [10], [11], [12], [13]. The common practice of discriminative training is not generally feasible, however, making it especially challenging to describe the cross-domain relationship.

To handle the covariate shift, early domain adaptation works compute the probability of each sample belonging to the source or target domain via likelihood ratio estimation. One favorite principle for instance re-weighting is Maximum Mean Discrepancy (MMD) in two-sample statistical test [14]. Such instance re-weighting strategies are intuitive, however, they are always separated from the classifier training procedure. To deal with this drawback, Chu et al. [3] propose to jointly re-weight the training samples and learn a classifier. Meanwhile, Long et al. [9] learn a domain-invariant projection while both the conditional distribution and marginal distribution divergences are minimized. Due to the lack in labeled target data, the pseudo labels on target data and the projection function are optimized alternately. Baktashmotlagh et al. [15] investigate the Gaussian kernel into MMD, and minimize the within-class variance that encourages class clustering in source domain simultaneously.

Generally speaking, bridging the gap between the source and target domain and preserving the discriminative power for the labeled source data are two critical components for unsupervised DA methods. Some works [9], [10], [16] experimentally have proved that the pseudo labels of target data involved in the optimization process can significantly boost the adaptation performance. Specifically, the pseudo labels are exploited to minimize the empirical conditional distribution divergences (i.e., the differences between class-wise means), nevertheless, none of them have ever investigated the class clustering objective for the target domain. Therefore, we investigate a novel domain-irrelevant class clustering objective that is theoretically related to both the distribution divergence and the variance minimization terms involved in both domains. To illustrate the necessity of the target clustering structure, we also provide a toy example in Fig. 1, where the differences

• *The authors are with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition (NLPR), CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences (CASIA) and University of Chinese Academy of Sciences, Beijing 100190, China. Email: {jian.liang, rhe, znsun, tnt}@nlpr.ia.ac.cn.*

between class-wise means are minimized in both subspaces. Clearly, in the left subspace, the classification performance is expected to be much worse than it in the right one due to its small margin for projected target data. The proposed objective is further decomposed into three terms, i.e., the empirical conditional distribution divergence and two intra-domain class-clustering terms. We naturally obtain a relaxed domain-irrelevant class-clustering objective by introducing a balancing parameter for the intra-domain terms. Obviously, to infer the optimal projection, we still need to know the pseudo target labels. As such, we jointly learn the domain-invariant projection via the proposed objective and infer the pseudo target labels in a loop. In each iteration, the pseudo labels are inferred via the classifier trained on the projected source data.

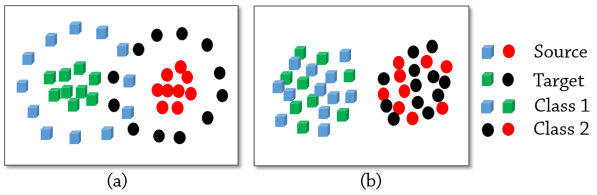


Fig. 1. A toy example of the sought subspaces with different structures, (a). promoting source class-clustering, (b). promoting domain-irrelevant class-clustering (confusion), respectively. (Best viewed in colors.)

Apart from the domain-invariant projection inference, how to label target data is also a critical issue for unsupervised domain adaptation. Nevertheless, this issue has long been overlooked, and a classifier is usually trained on the whole projected source data to classify the projected target data. Classifiers trained in this manner are likely to be overfitting for a homogeneous task, let alone the heterogeneous task on the projected source and target domains. Benefiting from the combination of multiple feature representations, previous works [4], [17] achieve promising performance in domain adaptation. Inspired by these methods and classical ensemble models such as bootstrap aggregating (bagging) [18], we further exploit several coupled source-target sub-domain pairs to learn various local domain-invariant projection functions. To further increase the robustness, we also adopt the idea of random feature selection in random forest [18] that has been proven to be less overfitting. Hence, the original problem has been distributed into many small-sized problems, which decreases the time complexity dramatically and is also desirable for unsupervised DA tasks with large-scale instances and high-dimensional features.

Towards this end, we first propose a novel domain-invariant projection ensemble framework for unsupervised domain adaptation. Concerning the domain-invariant projection, we propose a novel objective function that considers both the conditional distribution divergence and class clustering promoting terms involved in both domains. To further enhance the performance, a 'sampling-and-fusion' strategy in Fig. 2 is developed to improve the generalization ability. Especially, to infer the domain-invariant projection for each cross-domain subset, we arrive at a generalized eigenvalue decomposition problem, which has a closed-form solution. To label the target data in a loop, we adopt a Nearest Neighbor (NN) classifier on the

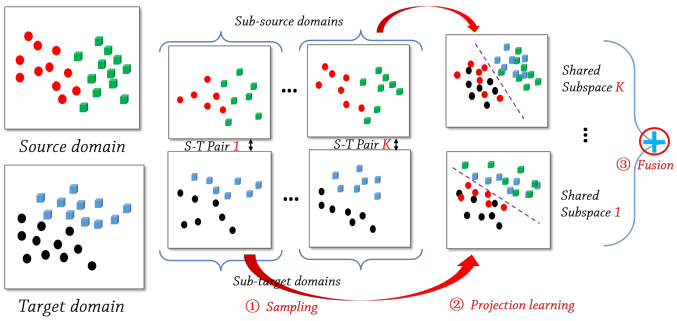


Fig. 2. Overview of the proposed 'sampling-and-fusion' framework. Several coupled source-target domain subsets are generated via randomly instances and feature sampling. Then for each source-target domain pair, we learn the optimal domain-invariant projection via the proposed domain-invariant object function. Finally these classifiers are trained on the whole source data and be fused in an ensemble manner to predict the whole unlabeled target data. (Best viewed in colors.)

projected subspace for simplicity. Finally, we provide two popular strategies, i.e., feature concatenation and majority-voting scheme in the fusion step. Overall, the problem is also computationally efficient, and is flexible to large-scale unsupervised domain adaptation. The contributions of this paper are summarized as follows.

- To compensate for the lack of target structure constraint, we propose a novel objective function relaxed from a Domain-Irrelevant Class clustEring (DICE) term for unsupervised domain adaptation. The optimal projection and pseudo target labels are alternately optimized, and in each iteration the projection is computed in closed-form via solving a generalized eigenvalue problem.
- An ensemble strategy is first exploited for unsupervised domain adaptation problems, where we construct various domain adaptation tasks via randomly selecting the instances and identical features for both domains and infer the corresponding domain-invariant projections, making the ensemble method faster and trivially parallelizable in the projection inference step.
- Extensive empirical experimental results on several benchmark datasets demonstrate that the proposed methods achieve performances superior to state-of-the-art unsupervised domain adaptation methods and the ensemble method always outperforms its single-projection one with noticeable margins. Particularly, on the challenging cross-view PIE database, DICE advances the best accuracies from 58.8% [19], 65.1% [20] to 80.6%. When combined with deep features, DICE is even competitive with current state-of-the-art deep methods.

The remainder of this paper is organized as follows. In Section 2, we provide a brief review on previous unsupervised domain adaptation methods. Section 3 is dedicated to introducing some background knowledge. In Section 4, we introduce the proposed methods with the novel objective formulation and some sampling-and-fusion strategies. Section 5 is devoted to the experimental setting, comparisons, and analysis, which demonstrate the superior performance of our methods. Section 6 concludes this paper with possible future research directions.

2 Prior Work

In this section, we mainly review previous unsupervised domain adaptation methods and divide them into two main categories: shallow methods (e.g., instance re-weighting, feature augmentation, and feature transformation methods) and end-to-end deep adaptation methods.

Despite the distribution difference among domains, instance re-weighting methods usually assume the identical conditional distributions. Maximum entropy density estimation is exploited to infer the re-sampling weights [21], while [22] estimates the re-sampling weights by matching training and test distribution feature means in a reproducing kernel Hilbert space, i.e., Maximum Mean Discrepancy (MMD). Besides optimizing the weights, [3] also learns the optimal parameters of a classifier.

One incredibly simple approach [7] firstly augments the features for both domains, then adopts the trained classifier on source domain to predict unknown target data, this easy strategy even beats state-of-the-art approaches at that time. [23] extends [7] by utilizing two newly proposed feature mapping functions for heterogeneous domain adaptation. Besides, [2], [4] propose to embed each domain into one d-dimensional linear subspace, subsequently, a geodesic path between the source and target domain is built. Then new feature representations are obtained via sampling points in the path and concatenating these intermediate domains. GFK [24] further defines a geodesic-flow kernel that integrating over all the intermediate subspaces lying on the geodesic path without sampling. Besides, [17] chooses different amounts of source samples from coarse to fine via MMD, and generates several new feature representations via GFK for each level. [25] proposes a pioneering semi-supervised cross-domain kernel learning framework which can incorporate many existing kernel methods.

Additionally, feature transformation that aligns source and target domains is a more natural choice for DA. SA [26] discovers an optimal transformation matrix to minimize the Bergman matrix divergence between two PCA subspaces. Recently, CORAL [27] investigates the second-order statistics instead of first-order MMD, which equips the target domain with the same covariance matrix of the source domain. [28] further utilizes the second-order and higher-order scatter tensor to learn the optimal transformation. Besides direct alignment transformations, another popular paradigm attempts to infer domain-invariant features via dimensionality reduction. Following this idea, [29] learns a transformation matrix that minimizes the distance between the source and target domains via MMD and preserves the data variances. JDA [9] firstly considers the conditional distribution, where the class means besides the total means are also required to be close to each other. Taking into consideration both the supervised source class clustering and the inter-domain MMD term, [15] further formulates this objective with an orthogonal constraint as an optimization problem on the Grassmann manifold. The Hellinger distance and polynomial kernel are investigated to increase the robustness of manifold-based DA methods in [5]. [11] discovers the projection via inter-domain second-order information on the SPD manifold. [16] further extends [9] to address that the inter-domain class means belonging to different classes should be pushed far away. By contrast, JGSA [10] learns two different transformation matrices for each domain, thus, a subspace alignment constraint in [26] is further developed to be combined with other statistical alignment objectives. SCA [30] takes the between and within class scatters of the source domain into consideration.

More recently, deep learning techniques have achieved remarkable successes for computer vision [31], [32]. Benefiting from the powerful deep neural networks, [33], [34], [35], [36], [37], [38] obtain promising classification performance on several benchmark domain adaptation datasets. [35] extends [27] by leveraging the second-order correlation alignment loss into the deep framework. [34] considers the multi-kernel MMD defined among several layers, while [33] merely utilizes linear MMD on a single layer. Furthermore, [36] focuses on the joint distribution discrepancies instead of the marginal one. [39] describes an end-to-end deep learning framework for jointly optimizing the optimal deep feature representation, cross domain transformation, and the target label inference for state-of-the-art unsupervised domain adaptation. Besides these discrepancy-based methods, adversarial loss functions are also favored in deep domain adaptation methods [40], [41], [42], [43]. Generally, adversarial models aim to introduce a novel domain discriminator to promote domain confusion, namely, this discriminator cannot determine which domain the data come from. In this manner, these two domains are considered to be drawn from the same distribution [44]. Existing adversarial DA methods attempt to jointly reduce the domain divergence and preserve the discriminative ability of source data, and a summarization of these methods can be referred in [43]. Regarding the loss function, [42], [43], [45] exploit the min-max loss, inverted label GAN loss and confusion loss, respectively. By contrast, [46] proposes to learn a joint distribution of multi-domain images with a weight-sharing constraint on generators.

Apart from the aforementioned feature-level methods, [47] only exploits the black-box source classifier learned on source domain to preserve the privacy of source data. Although most existing DA methods are proposed for homogeneous transfer learning problems, there are still some approaches that leverage feature augmentation [23] or learn an intermediate domain [48], [49] to bridge the gap across domains for heterogeneous DA. In addition, webly-supervised DA methods [50], [51] extract privileged information from freely available web videos for action and event recognition, which is a fertile research direction.

Generally, despite the promising performance, there are some limitations for these shallow methods, 1). the target structure is usually ignored in adaptation models; 2). they can not cope well with different label distributions within two domains. Besides, deep domain adaptation methods not only rely on the high-capability computers, but also enjoy a relatively long training time as well as a challenging parameter-tuning process. To this end, we propose a novel domain discrepancy objective to consider the clustering structure in target domain for domain-invariant projection inference, and introduce an ensemble framework to increase its classifier's discriminative ability and reduce the complexity simultaneously, making it even flexible for large-scale high-dimensional datasets.

3 NOTATIONS AND PRELIMINARIES

Let $\mathcal{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ denote n_s data points and their associated labels of the source domain. Likewise, we denote $\mathcal{D}_t = \{(x_t^j, y_t^j)\}_{j=1}^{n_t}$ n_t data points of the target domain. For unsupervised domain adaptation, samples like x_s^i and x_t^j share the same feature dimensionality d, $y_t^j \in \{0,1\}^C$ is not known in the training phrase, and C is the number of classes. For simplicity, $X_s \in \mathbb{R}^{d\times n_s}$ and $X_t \in \mathbb{R}^{d\times n_t}$ indicate all the source and target data respectively, and each column of $X_{s/t}$ represents a data point in $\mathcal{D}_{s/t}$. Besides, $Y_s \in \{0,1\}^{n_s\times C}$ and $Y_t \in \{0,1\}^{n_t\times C}$ denote the one-hot encodings with the semantic information, while $y_s^i(a) = 1$ means that i-th source data is associated with the a-th class. The domain-invariant projection function is defined as simple linear function $f(x) = A^T x$, and the projection parameter $A \in \mathbb{R}^{d\times m}$, where m is the subspace dimensionality. $\|A\|_F := \sqrt{tr(AA^T)}$ represents the Frobenius norm of A, where $tr(\cdot)$ denotes the trace of a square matrix, and $\|a\|_2 = \sqrt{a^T a}$ denotes the l_2 norm of one column vector a. $\mathbf{I}$ denotes the identity matrix, and $\mathbf{1}$ is the vector of all ones with appropriate dimensionality, and $\mathcal{H}$ is the Reproducing Kernel Hilbert Spaces (RKHS).

As stated above, one intuitive and effective solution for domain adaptation is to seek one projection function, namely domain-invariant projection (DIP), via which different domains nearly share the same distribution. Among these DIP methods, JDA [9] is a classical one that tries to discover a projection function that adapts **joint distributions** including both marginal and conditional distributions between domains without any labeled target data. The mathematical formulation for the joint distribution difference, which can be further decomposed into two different distribution differences, is shown below:

$$\begin{aligned}\min_T & \underbrace{\left\|\mathbb{E}_{P(x_s,y_s)}\left[T\left(x_s\right), y_s\right] - \mathbb{E}_{P(x_t,y_t)}\left[T\left(x_t\right), y_t\right]\right\|^2}_{\text{joint distribution difference}} \\ & \approx \underbrace{\left\|\mathbb{E}_{P_s(x_s)}\left[T\left(x_s\right)\right] - \mathbb{E}_{P_t(x_t)}\left[T\left(x_t\right)\right]\right\|^2}_{\text{marginal distribution difference}} \\ & + \underbrace{\left\|\mathbb{E}_{Q_s(y_s|x_s)}\left[y_s|T\left(x_s\right)\right] - \mathbb{E}_{Q_t(y_t|x_t)}\left[y_t|T\left(x_t\right)\right]\right\|^2}_{\text{conditional distribution difference}}\end{aligned} \tag{1}$$

, where $T(\cdot)$ is the optimal projection function to be sought, and $\mathbb{E}_p(f(x))$ is the expectation of $f(x)$ under p.

For such distribution differences involved in domain adaptation, Maximum Mean Discrepancy (MMD) [14] has been a widely used measure tool. MMD is primarily proposed to tackle the two-sample problem (samples come from two probability distribution p and q) via a statistical test of the hypothesis that these two distributions are different. The main idea of MMD is to find a smooth function via which the difference between the mean function values (namely, mean discrepancy) on p and q is largest. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$, then the expressions of MMD and its biased empirical estimate are defined as

$$\begin{aligned} MMD[\mathcal{F}, p, q] &:= \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]\right), \\ MMD_b[\mathcal{F}, X, Y] &:= \sup_{f\in\mathcal{F}} \left(\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right). \end{aligned} \tag{2}$$

Here $X = \{x_i\}_{i=1}^m$ and $Y = \{y_i\}_{i=1}^n$ denote m and n data points i.i.d. sampled from p and q, respectively. From the definition above, we can see that $MMD = 0$ if and only if p is indistinguishable from q (namely, $p = q$). Next, when $\mathcal{F}$ becomes a kernel function set $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ in a universal RKHS, then the witness function and its biased empirical estimate can be referred in [52].

4 THE PROPOSED FRAMEWORK

4.1 Domain-invariant Projection via Domain-irrelevant Class Clustering

Even most efforts have been devoted to minimizing the distribution difference between the source and target domain, how to leverage the semantic information in the labeled source domain is also critical for unsupervised DA methods. While some approaches [26], [27] totally split projection learning and classifier learning, they merely exploit the semantic labels to the final classifier training phrase after alignment or projection, thereby obtaining inferior performance. As shown in [10], [15], a class-clustering promoting term is utilized to increase the compactness of each class and preserve the discrimination of the projected source domain. Generally, the overall objective function of DIP that includes the class clustering promoting term is formulated as below:

$$A^* = \arg\min_{A\in\mathbb{S}} \Phi(A^T X_s, A^T X_t) + \lambda \sum_{c=1}^{C}\sum_{i=1}^{n_s^c} \|A^T(x_s^i - \mu_{s,c})\|_2^2. \tag{3}$$

Here function $\Phi(\cdot,\cdot)$ tries to align the source and target domain, and $\mu_{s,c} = \frac{\sum_{y_s^i(c)=1} x_s^i}{\sum_i y_s^i(c)}$ is the c-th class center in the source domain, and $\mathbb{S}$ is a possible constraint on the projection function A, e.g., orthogonality constraint $S = \{A|A^TA = I\}$ in [15].

Besides, inspired by recent works [9], [14], we can simply use the empirical MMD in Eq. (2) as the distribution difference measure to compare different distributions, i.e., the distance between the sample means of source and target data. Hence in this manner, Eq. (1) can be simplified as,

$$\min_A \Gamma(A) = \|A^T(\mu_s - \mu_t)\|_2^2 + \sum_{c=1}^{C} \|A^T(\mu_{s,c} - \mu_{t,c})\|_2^2, \tag{4}$$

where the first term is the marginal distribution difference measured via MMD and the second term represents the conditional distribution difference likewise. Besides, $\mu_s = \frac{\sum_i x_s^i}{n_s}$ and $\mu_t = \frac{\sum_i x_t^i}{n_t}$ are the global centers of source and target data, respectively. Likewise, $\mu_{t,c} = \frac{\sum_{\hat{y}_t^i(c)=1} x_s^i}{\sum_i \hat{y}_t^i(c)}$ is the c-th class cluster in the target domain, $\hat{y}_t \in \{0,1\}^C$ is the pseudo label vector estimated from the previous iteration.

Although these DIP approaches achieve substantial gains over previous state-of-the-art methods, they do not even consider the same class-clustering encouraging loss on the target domain. On the contrary, JDA involves the pseudo labels on target domain into seeking an optimal projection, and experimentally proves the effectiveness of them for unsupervised DA. Besides, [53] investigates the discriminative clustering for target data, which effectively improves the adaption performance. Hence, we believe that encouraging class clustering regardless of the source

or target domain jointly with minimizing the distribution differences across domains can further boost the adaptation performance. Then in this paper, we propose a novel objective function that connects two domains together in terms of the projection A,

$$\min_A \Omega(A) = \sum_{c=1}^{C} \sum_{x \in D_c} \|A^T(x - \mu_c)\|_2^2, \tag{5}$$

where $D_c = \{x_s^i | y_s^i(c) = 1\} \cup \{x_t^j | \hat{y}_t^j(c) = 1\}$ consists of all data points that are associated with one identical class c from both source and target domains, and $\mu_c = \frac{\sum_{y_s^i(c)=1} x_s^i + \sum_{\hat{y}_t^j(c)=1} x_t^i}{\sum_i y_s^i(c) + \sum_i \hat{y}_t^i(c)}$ is the domain-irrelevant class center. Obviously, once the pseudo target labels are obtained, this term above will bring all data with the same class label together. That is to say, all data from the same class are close to each other, hence, both the intra-domain variance and the inter-domain differences are minimized at the same time. As shown in Fig. 1, it indeed increases the compactness of each class in both domains, and it seems that this domain-irrelevant clustering term can also pull closer the heterogeneous centers from same class in different domains.

To investigate the relation with conditional distribution difference in Eq. (4), denoting by $D_c^s = \{x_s^i | y_s^i(c) = 1\}$ and $D_c^t = \{x_t^j | \hat{y}_t^j(c) = 1\}$ samples from the c-th class in the source and target domain respectively, we further rewrite the objective function in Eq. (5) as follows. [$x \leftarrow A^T x$]

$$\begin{aligned}
&\sum_{x \in D_c} \|x - \mu_c\|_2^2 = \sum_{x \in D_c^s} \|x - \mu_c\|_2^2 + \sum_{x \in D_c^t} \|x - \mu_c\|_2^2 \\
&= \sum_{D_c^s} \|x - \mu_{s,c} + \mu_{s,c} - \mu_c\|_2^2 + \sum_{D_c^t} \|x - \mu_{t,c} + \mu_{t,c} - \mu_c\|_2^2 \\
&= \sum_{x \in D_c^s} \|x - \mu_{s,c}\|_2^2 + \sum_{x \in D_c^t} \|x - \mu_{t,c}\|_2^2 \\
&+ n_s^c \cdot \|\mu_{s,c} - \mu_c\|_2^2 + n_t^c \cdot \|\mu_{t,c} - \mu_c\|_2^2 \\
&= \sum_{x \in D_c^s} \|x - \mu_{s,c}\|_2^2 + n_s^c \|\mu_{s,c} - \frac{n_s^c \mu_{s,c} + n_t^c \mu_{t,c}}{n_s^c + n_t^c}\|_2^2 \\
&+ \sum_{x \in D_c^t} \|x - \mu_{t,c}\|_2^2 + n_t^c \|\mu_{t,c} - \frac{n_s^c \mu_{s,c} + n_t^c \mu_{t,c}}{n_s^c + n_t^c}\|_2^2 \\
&= \sum_{x \in D_c^s} \|x - \mu_{s,c}\|_2^2 + \sum_{x \in D_c^t} \|x - \mu_{t,c}\|_2^2 + \beta \|\mu_{s,c} - \mu_{t,c}\|_2^2.
\end{aligned} \tag{6}$$

Here the entire center of the c-th class over both domains is denoted by μ_c, and $n_s^c = \sum_i y_s^i(c)$ and $n_t^c = \sum_j \hat{y}_t^j(c)$ denote the size of the c-th source and target class. To help understand the deduction, the middle terms $\sum_{y_s^i(c)=1}(x - \mu_{s,c}) = 0$ and $\sum_{y_t^j(c)=1}(x - \mu_{t,c}) = 0$ are further discarded. Besides, the trade-off parameter $\beta = (\frac{n_s^c n_t^c}{n_s^c + n_t^c})$ is a constant. Furthermore, the data point x and its related center μ can be easily replaced with their projected expressions. That is to say, the domain-irrelevant class-clustering promoting term can be decomposed into three terms, i.e., conditional distribution difference in Eq. (4), source class-clustering promoting term in Eq. (3), and that of target domain. As far as we know, it is the first attempt to discover the effectiveness of variance minimization on target domain and analyze the relationship between the intra-domain variance minimization and the cross-domain empirical distribution difference minimization.

However, Eq. (5) favors larger-size categories with larger β. To allow for more flexibility and alleviate the bias toward the majority class, we introduce a unified balancing parameter λ instead of the size-dependent constant $1/\beta$ before the class-clustering encouraging terms, henceforth, the final objective function including the empirical margin distribution difference minimization is rewritten as

$$\begin{aligned}
\min_A \mathcal{L}(A) &= \|A^T(\mu_s - \mu_t)\|_2^2 + \sum_{c=1}^{C} \|A^T(\mu_{s,c} - \mu_{t,c})\|_2^2 \\
&+ \lambda \sum_{c=1}^{C} \left[\sum_{i=1}^{n_s^c} \|A^T(x_s^i - \mu_{s,c})\|_2^2 + \sum_{i=1}^{n_t^c} \|A^T(x_t^i - \mu_{t,c})\|_2^2 \right].
\end{aligned} \tag{7}$$

Interestingly, JDA is a special case of our proposed model when $\lambda = 0$, and DIP-CC [15] can be considered as a special case where only the first and third terms are considered.

4.2 Target Label Estimation via Projection Ensemble

As mentioned above, in this paper we propose a novel unsupervised DA method which alternately learns the domain-invariant projection and infers the pseudo target labels. We have clarified how to calculate the projection via the pseudo target labels $\hat{y}_t$ in Section 4.1, hence, we will explain how the pseudo labels are obtained after projection.

Once the source domain and the target domain are aligned via DIP, we can expect the commonly used classification models trained on the source domain, e.g., Support Vector Machine (SVM) and K-Nearest Neighbor classifier (KNN) to achieve considerable classification performance in the target domain. However, it is well known these classifiers are prone to be over-fitting when directly applied for the homogeneous testing dataset due to the covariate shift, let alone the heterogeneous projected testing dataset. Inspired by the idea of ensemble methods such as bagging and random forest [18], we randomly sample couple subsets of both domains even features, and form some sub-Source-Target domain pairs (see Fig. 2). Apart from the over-fitting effect, the pseudo target label is involved into the training procedure, hence making it critical to obtain better initialization. Learning with multiple random sub-Source-Target domain pairs can yet generate various domain-invariant projections, the fusion of these 'local' projections rather than the 'global' sought projection is expected to enhance the DA performance. *Note that the training process of each subset of coupled domains is independent, namely, we seek each optimal domain-invariant projection for each coupled subset in parallel*. However, this manner is quite different from multi-source domain generalization [54] where classifiers from different source are fused. In the following we will provide several fusion strategies to infer unlabeled target data.

Denote by $\{\widehat{X}_s^{(i)}\}_{i=1}^K$ and $\{\widehat{X}_t^{(i)}\}_{i=1}^K$ the K sampled coupled source-target domain pairs respectively, and $\widehat{A}^{(i)} \in \mathbb{R}^{d \times m}$ is the optimal projection learned on the i-th source sub-domain $\widehat{X}_s^{(i)}$ and target sub-domain $\widehat{X}_t^{(i)}$. For simplicity, we further denote $Z_s^{(i)}$ and $Z_t^{(i)}$ as the embedded features of complete source and target domains via the i-th embedding projection $\widehat{A}^{(i)}$ below ($i \in [1, K]$): $Z_s^{(i)} = \widehat{A}^{(i)T} X_s, Z_t^{(i)} = \widehat{A}^{(i)T} X_t$.

Majority Voting (MV) Benefiting from the random sampling strategy and following voting rules, the bagging

algorithm can achieve promising performance with lower variance. Here we can follow the idea of bagging methods, namely, after we have obtained K predictors/classifiers trained on each low-dimensional source data $Z_s^{(i)}$, the most prevalent fusion rule, i.e., majority voting, is directly exploited to decide the final class of each target data. Concretely, each classifier $f_i^r : \mathbb{R}^m \rightarrow \mathbb{R}$ calculates the probability scores of the r-th class for the i-th projected Source-Target pair. Then the votes casted by all K classifiers are then counted and the candidate class which achieves the maximum number of votes is considered as the final class $\bar{y}$ of one target domain data point $x \in \mathbb{R}^d$,

$$\bar{y} = \arg\max_e \sum\nolimits_{i=1}^{K} \sigma_e(s_i), \tag{8}$$

$$\text{where } s_i = \arg\max_r f_i^r(z_t^{(i)}),\ \sigma_e(s_i) = \begin{cases} 1, & s_i = e, \\ 0, & \text{otherwise.} \end{cases}$$

Weighted Majority Voting (WMV) Each classifier in MV only votes for one class, namely the votes for other classes equal zero, which may drop some information. WMV considers the classification scores as the continuous weighted votes $f_i^t(x)$ instead of the discrete $\{\mathbf{0,1}\}$ votes, hence the rule of WMV is defined as

$$\bar{y}_i = \arg\max_e \sum\nolimits_{i=1}^{K} f_i^e(x). \tag{9}$$

As such, classifiers that can output probability scores should be chosen. Else wise, a classifier like 1-NN can only generate discrete scores, making that WMV equals to MV in Eq. (8).

Feature-level Fusion (FF) By contrast, FF combines different projections $\hat{A}^{(i)}$ instead of classifiers together, which resembles the feature concatenation methods. That is to say, we can concatenate all learned domain-invariant projections to obtain $A_{con} = [\hat{A}^{(1)}, \cdots, \hat{A}^{(K)}] \in \mathbb{R}^{d \times Kc}$. Then we can train a classifier on the source domain $A_{con}^T X_s$, and directly predict on the target domain $A_{con}^T X_t$.

While we adopt the Euclidean distance based NN classifier for example, the feature concatenation projection A_{con} can be understood from a different perspective,

$$\|A_{con}^T(x_1 - x_2)\|_2^2 = \sum\nolimits_{i=1}^{K} \|\hat{A}^{(i)T}(x_1 - x_2)\|_2^2, \tag{10}$$

for every two samples $x_1, x_2 \in \mathbb{R}^d$. Namely, the distances from each projection are cast up to form the total distance for any paired data points. Likewise, once the features are normalized to unit length, the cosine distance based NN classifier owns the same property.

4.3 Reformulation & Optimization

The proposed objective function $\mathcal{L}(A)$ in Eq. (7) is not intuitive to solve, hence we introduce several variables $Q^0 \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ and $Q^c,\ c \in [1, C]$ to simplify the optimization problem, each element is defined as

$$Q_{ij}^0 = \begin{cases} \frac{1}{n_s n_s},\ x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t},\ x_i, x_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t}, \text{otherwise.} \end{cases}, Q_{ij}^c = \begin{cases} \frac{1}{n_s^c n_s^c},\ x_i, x_j \in \mathcal{D}_s^c \\ \frac{1}{n_t^c n_t^c},\ x_i, x_j \in \mathcal{D}_t^c \\ \frac{-1}{n_s^c n_t^c},\ x_i \in \mathcal{D}_s^c, x_j \in \mathcal{D}_t^c \\ \frac{-1}{n_s^c n_t^c},\ x_j \in \mathcal{D}_s^c, x_i \in \mathcal{D}_t^c \\ 0, \quad \text{otherwise.} \end{cases} \tag{11}$$

Here x_i and x_j are the i-th and j-th columns of $X = [X_s, X_t] \in \mathbb{R}^{d\times(n_s+n_t)}$, respectively. In this manner, the first and second terms of Eq. (7) can be rewritten as

$$\left\| A^T \left(\frac{1}{n_s} \sum_{i=1}^{n_s} x_i - \frac{1}{n_t} \sum_{j=n_s+1}^{n_s+n_t} x_j \right) \right\|_2^2 = tr\left(A^T X Q^0 X^T A \right), \tag{12}$$

where Q^0 is also referred as the MMD matrix. Then the second term in Eq. (7) can be rewritten as

$$\left\| A^T \left(\sum_{i=1}^{n_s} x_i y_i(c)/n_s^c - \sum_{j=n_s+1}^{n_s+n_t} x_j y_j(c)/n_t^c \right) \right\|_2^2 = tr\left(A^T X Q^c X^T A \right),$$

where $Y = [Y_s; \hat{Y}_t] \in \{0,1\}^{(n_s+n_t)\times C}$ is the whole one-hot encoding of semantic labels.

Inspired by Linear Discriminant Analysis (LDA), we further transform the last two class-clustering promoting terms that can be considered as within-class variance in Eq. (7) into similar expressions as Eq. (12).

$$\begin{aligned} &\sum_{c=1}^{C} \sum_{i=1}^{n_s^c} \|A^T(x_s^i - \mu_{s,c})\|_2^2 = \|A^T(X_s - X_s Y_s (Y_s^T Y_s)^{-1} Y_s^T)\|_F^2 \\ &= tr(A^T X_s (I - Y_s (Y_s^T Y_s)^{-1} Y_s^T) X_s^T A), \\ &\sum_{c=1}^{C} \sum_{i=1}^{n_t^c} \|A^T(x_t^i - \mu_{t,c})\|_2^2 = tr(A^T X_t (I - \hat{Y}_t (\hat{Y}_t^T \hat{Y}_t)^{-1} \hat{Y}_t^T) X_t^T A). \end{aligned}$$

Then we can naturally combine these two terms above together into a new variable $Q^{cc} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ as

$$Q^{cc} = \begin{bmatrix} I - Y_s (Y_s^T Y_s)^{-1} Y_s & \mathbf{0} \\ \mathbf{0} & I - \hat{Y}_t (\hat{Y}_t^T \hat{Y}_t)^{-1} \hat{Y}_t \end{bmatrix}. \tag{13}$$

To this end, we have provided the reformulations of each term in Eq. (7), thus it is obvious to obtain the following equivalent objective function $\Psi(A) = tr(A^T X Q^{all} X^T A)$, where $Q^{all} = \sum_{c=0}^{C} Q^c + \lambda Q^{cc}$, with both terms being normalized to unit F-norm. To avoid a non-trivial solution, we further consider to maximize the variances for both domains, namely $tr(A^T X H X^T A)$, where $H = I - \frac{1}{n_s+n_t}\mathbf{1}$ is the centering matrix. Therefore, the overall optimization problem w.r.t. A reduces to the following formulation

$$\begin{aligned} &\min_A tr(A^T X Q^{all} X^T A) + \gamma \|A\|_F^2, \\ &s.t.\ A^T X H X^T A = I, \end{aligned} \tag{14}$$

where γ is a hyper-parameter to avoid numerical instability issue. Interestingly, the optimal solution of $A \in \mathbb{R}^{d\times m}$ that satisfies the above objective function is given by the generalized eigenvalue problem:

$$(X Q^{all} X^T + \gamma I) a = \eta X H X^T a, \tag{15}$$

where η is the i-th minimum eigenvalues and $a \in \mathbb{R}^d$ is the associated i-th column in A, $i \in [1, m]$.

Kernel extension Note that we only utilize a linear projection $A^T X$ for domain adaptation, since some studies show the superiority of non-linear functions, we further consider the non-linear kernel-mapping $\psi : x \rightarrow \psi(x)$ to enhance the adaptation ability. Then for the kernel matrix $\mathcal{K} = \psi(X)^T \psi(X) \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$, we exploit the Representer theorem [55] and obtain the following problem,

$$\begin{aligned} &\min_{A_k} tr(A_k^T \mathcal{K} Q^{all} \mathcal{K}^T A_k) + \gamma \|A_k\|_F^2, \\ &s.t.\ A_k^T \mathcal{K} H \mathcal{K}^T A_k = I, \end{aligned} \tag{16}$$

where $A_k^T K_e \leftarrow (\psi(X)A)^T\psi(X)$. The solutions to both Eq. (14) and Eq. (16) are quite easy to implement.

Label Propagation extension Taking into consideration the global structure in the pseudo label inference step, we further leverage a desirable closed-form label propagation (LP) method [20], [56] after the 1-NN classifier. Please refer to the supplementary materials for more details.

Apart from these extensions, we also propose a bagging-like unsupervised DA algorithm in this paper, X_s and X_t should be replaced by their respective subsets $X_s^{(i)}$ and $X_t^{(i)}$ to seek the optimal projection $A^{(i)}$ via Eq. (15). Once we obtain all K domain-invariant projections based on different subsets, we resort to these fusion strategies (i.e., MV, WMV and FF) in section 4.2 to build a more robust and accurate predictor for target data. The pseudo code for our proposed method is summarized in Algorithm 2.

Algorithm 1 Basic domain-invariant projection method

Input: Source data $\{\mathcal{X}_s, Y_s\}$ and target data $\mathcal{X}_t$; subspace dimensionality m, trade-off parameters λ, γ.
Output: Domain-invariant projection A.
1: $X \leftarrow [X_s, X_t]$
2: Construct the MMD matrix Q^0 in Eq. (11);
3: Initiate $Q^{all} = Q^0$ due to the lack in $\hat{Y}_t$;
4: **repeat**
5: Obtain optimal A via solving the generalized eigenvalue decomposition problem in Eq. (14);
6: Train a standard classifier f on $\{A^T X_s, Y_s\}$ and update the pseudo labels $\hat{Y}_t$ for target data $A^T X_t$;
7: Build the class-wise MMD matrices $\{Q^c\}_{c=1}^C$ and the class-clustering matrix Q^{cc} in Eq. (11) and Eq. (13), then normalize them;
8: $Q^{all} \leftarrow \sum_{c=0}^C Q^c + \lambda Q^{cc}$;
9: **until** Convergence or maximum iteration achieved
10: Train a classifier f on $\{A^T\mathcal{X}_s, Y_s\}$.

Algorithm 2 Domain-invariant projection ensemble method

Input: Source data $\{\mathcal{X}_s, Y_s\}$ and target data $\mathcal{X}_t$; # sub-domain-pairs K, subspace dimensionality m, parameter λ; sampling density $\delta_{s,t,f}$.
Output: Projection matrices $\{\hat{A}^{(i)}\}_{i=1}^K$, final classifier f.
1: **for** $i = 1 : K$ **do**
2: Uniformly sampling $X_s^{(i)}, Y_s^{(i)}$ from $\mathcal{X}_s$ at density δ_s;
3: Uniformly sampling $X_t^{(i)}$ from $\mathcal{X}_t$ at density δ_t;
4: Uniformly sampling feature subsets f from $[1 : d]$ at density δ_f;
5: $X_s = X_s(f, :), X_t = X_t(f, :)$
6: Obtain projection $\hat{A}^{(i)}$ via Algorithm 1;
7: **end for**
8: Retrain each classifier f_i on $\{A^{(i)T}\mathcal{X}_s, Y_s\}$, $i \in [1, K]$;
9: **switch** fusion method **do**
10: **case** (MV) Obtain the prediction probability scores and estimate the target label via Eq. (8);
11: **case** (WMV) Obtain the prediction probability scores and estimate the target label via Eq. (9);
12: **case** (FF/Con-) Concatenate these projections via Eq. (10), and train a new classifier on $\{A_{con}^T\mathcal{X}_s, Y_s\}$.

4.4 Complexity Analysis

We analyze the proposed basic domain-invariant projection in Algorithm 1 and its projection ensemble method in Algorithm 2. Regarding the basic one, it consists of two main parts, projection inference and the nearest neighbor classification, within T iterations. Concretely, the projection inference step occupies $\mathcal{O}(md^2)$, and the classification step occupies $\mathcal{O}(mn_sn_t)$, and building the MMD matrices $Q^c(0 \leq c \leq C)$ occupies $\mathcal{O}((n_s + n_t)^2)$, and the remaining steps like matrix multiplication occupy $\mathcal{O}(md(ns + nt))$. Thus, the overall time complexity of Algorithm 1 is $\mathcal{O}(Tmd^2 + Tmn_sn_t + T(n_s + n_t)^2 + Tmd(n_s + n_t))$.

Denote by $\delta_f, \delta_s, \delta_t$ the sampling densities for the feature, source and target instances, respectively. Obviously, the time complexity of basic method decreases to $\mathcal{O}(Tm\delta_f^2d^2 + Tm\delta_s\delta_tn_sn_t + T(\delta_sn_s + \delta_tn_t)^2 + Tm\delta_fd(\delta_sn_s + \delta_tn_t))$. After obtaining K projections, taking the MV strategy for instance, the final classification step occupies $\mathcal{O}(Kmn_sn_t)$. Assume $\check{n} = \delta_sn_s + \delta_tn_t$, then the overall computation complexity of Algorithm 2 is $\mathcal{O}(KTm\delta_f^2d^2 + KTm\check{n}^2 + KT\check{n}^2 + Km\delta_fd\check{n})$.

4.5 Hyper-Parameter Settings

Before reporting the detailed evaluation results, it is vital to explain how DICE hyper-parameters are tuned. Empirically, λ is fixed at 1 for balancing the inter-domain and intra-domain objectives, thus, only two hyper-parameters γ, m remain tunable. Since no target labels are available for unsupervised DA, it is impossible to conduct a standard cross-validation. Hence we perform k-fold cross-validation on the labeled source domain, namely, calculating the averaged accuracy on each one source fold while exploiting the other $k-1$ source folds and the whole target domain for training. In this manner, we obtain the optimal parameters $\gamma \in [0.001, 0.005, 0.01, 0.05, 0.1, 1, 5]$, m around the integral multiplies of C (# classes) via which the averaged source accuracy is the highest. Generally, this strategy is always sufficient to produce good DICE models for unsupervised DA. Similarly, we adopt the same strategy for finding the optimal parameters for ensemble methods.

5 Experiments

In this section, we comprehensively compare our methods with state-of-the-art unsupervised domain adaptation approaches on six visual benchmark datasets including object, face, and digit images. Details about parameter sensitivity are also discussed in the later part of this section.

5.1 Databases

□ **Office31** [57] includes images of 31 objects taken from 3 domains, i.e., Amazon (images downloaded from the online web merchants), DSLR (high-resolution images captured by a digital SLR camera) and Webcam (low-resolution images recorded by a web camera). Following [27], we exploit the AlexNet-FC_7 feature fine-tuned on the source domain.

□ **Office-Caltech** contains images from 10 overlapping object classes between the Office31 and Caltech256 [58] datasets. Previously, the SURF features [1] are adopted and encoded with 800-dimension BoW features. Besides, we also exploit $DeCAF_6$ features [10] and VGG-$FC_{6,7}$ features [11].

□ **Office-Home** [59] is a new benchmark dataset that contains 4 domains, with each domain containing 65 kinds of everyday objects, i.e., Art (artistic depictions of objects), Clipart (clipart images), Product (object without a background) and Real-World (object captured with a regular camera). We exploit the ResNet [32] to extract the features.

□ **PIE** [60] includes facial images of 68 people with various pose, illumination, and expression changes. Following [9], we mainly focus on 5 out of 13 poses, i.e., C05 (left), C07 (upward), C09 (downward), C27 (frontal) and C29 (right). These facial images are then cropped to the size 32×32.

1. Available at https://cs.stanford.edu/~jhoffman/domainadapt/

□ **MNIST-USPS** consists of two classical handwritten digit image datasets, USPS [61] and MNIST [62]. To speed up the experimental comparisons, we follow [9] to randomly choose 1,800 and 2,000 digit images from USPS and MNIST, respectively. Besides, the images from MNIST are uniformly resized to 16 × 16 to be consistent with USPS.

□ **COIL20** [63] is another object dataset that contains 1,440 samples over 20 classes with the image size 32 × 32. We split the dataset into two subsets COIL1 and COIL2 according to the capture directions [9]. Specifically, COIL1 contains all the images captured in the directions of $[0°, 85°] \cup [180°, 265°]$ while COIL2 contains remaining directions. Detailed information including all the dataset size and feature length is summarized in Table 1.

TABLE 1
Statistics of the six benchmark domain-adaptation datasets.

Dataset	Subsets	Abbr.	# Images	Feature (size)	# Classes
Office31	Amazon	A	2,817	AlexNet-FC$_7$(4,096)	31
	DSLR	D	498		
	Webcam	W	795		
Office-Caltech	Amazon	A	958	SURF (800)	10
	Caltech	C	1,123	DeCAF$_6$(4,096)	
	DSLR	D	157	VGG-FC$_6$(4,096)	
	Webcam	W	295	VGG-FC$_7$(4,096)	
Office-Home	Art	Ar	2,421		65
	Clipart	Cl	4,379	ResNet50-P$_5$(2,048)	
	Product	Pr	4,428	ResNet152-P$_5$(2,048)	
	Real-World	Rw	4,357		
PIE	C05 (←)	P1	3,332	Pixel (1,024)	68
	C07 (↑)	P2	1,629		
	C09 (↓)	P3	1,632		
	C27 (⊙)	P4	3,329		
	C29 (→)	P5	1,632		
MNIST-USPS	MNIST	M	2,000	Pixel (256)	10
	USPS	U	1,800		
COIL20	COIL1	C1	720	Pixel (1,024)	20
	COIL2	C2	720		

5.2 Baseline Methods & Experimental Setting

We compare our methods [2] with massive unsupervised domain adaptation methods [9], [10], [11], [12], [15], [16], [20], [24], [26], [27], [33], [34], [36], [37], [39], [42], [64], [65], [66], which can be summarized into three categories below: **1-NN based shallow methods**: 1-NN, GFK and GFK-pls [24] (here we only report the better model), SA [26], JDA [9], DIP-CC [15], CDDA [16], ILS [11], JGSA [10], OTGL [64], JDOT [65] and ATI [66]; **Non-1-NN based shallow methods**: SVM [3], SA* [26], CORAL [27] and DGA-DA [20]; **Deep methods**: DDC [33], DAN [34], DANN [42], DRCN [37], RTN [12], kNN-Ad [39], JAN [36], ADDA [43], and AutoDIAL [13]. Specifically, we re-run the public codes of GFK, SA, JDA, DIP-CC, ILS, JGSA and CORAL, and we implement CDDA and DGA-DA by ourselves. For the rest methods, the originally reported results are collected from their corresponding papers if available. Besides, we run SA* and CORAL with the provided source codes via LIBSVM [4].

Regarding the basic DICE, we adopt the 1-NN classifier during the training procedure due to its simplicity and parameter-free property. Apart from the primal DICE, we follow [9], [10] to develop two typical kernelized versions DICE$_{\text{lin}}$, DICE$_{\text{rbf}}$ and compare them with the kernelized variants of JGSA. To compare with non-1-NN based shallow methods, we also introduce DICE$_{\text{svm}}$ that differs from DICE in the last step where an overall SVM classifier instead of 1-NN classifier is utilized to predict the final target labels. Additionally, we enhance DICE with label propagation in the pseudo label inference step, i.e., DICE$_{\text{lp}}$, and compare it with DGA-DA. Note that all these methods are based on Algorithm 1 without the 'sampling-and-fusion' strategy.

Training Protocol For all the datasets in Table 1 except Office-Caltech, unless specified otherwise, we exploit all the source instances for training like [8], [9], [17]. Regarding the Office-Caltech dataset, an additional 'splitting' protocol is also adopted, which picks few source instances for each class for training that is adopted in previous works [2], [24], [26], [57], etc. Actually, we adopt the public 20 training splits in [67], [68], where 20 instances per class are selected for A and 8 for other domains. l_2 normalization is exploited on all features including pixel and deep features, moreover, we attempt z-score standardization for PIE in Table 5.

Parameter Setting In this paper, we always set λ to 1 and the maximum number of iterations to 10, while two parameters γ, m are selected through 5-fold cross-validation in Section 4.5. We also provide the optimal parameters here for different datasets, Office31 ($\gamma = 0.05$, $m = 30$), Office-Caltech ($\gamma = 0.1$, $m = 10$ for SURF & VGG-FC$_{6,7}$, $m = 15$ for DeCAF$_6$ & SURF with the 'splitting' protocol), Office-Home ($\gamma = 0.05$, $m = 100$), PIE ($\gamma = 0.01$, $m = 100$), MNIST-USPS ($\gamma = 0.1$, $m = 10, 30$) and COIL20 ($\gamma = 0.1$, $m = 10$). We include an experiment to show our solution's robustness to λ in Section 5.3.1.

TABLE 2
Recognition accuracies (%) on the **Office31** dataset.

type	method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.	Gain
1-NN	1-NN	55.2	50.6	41.2	94.8	40.8	98.6	63.5	–
	GFK-pls	58.2	59.4	45.9	95.6	43.8	98.6	66.9	5.4%
	SA	61.0	59.5	46.9	95.1	46.6	98.2	67.9	6.9%
	JDA	66.5	68.8	56.3	**97.7**	53.5	99.6	73.7	16.1%
	DIP-CC	56.0	51.9	44.0	95.3	42.3	98.8	64.7	1.9%
	CDDA	64.1	65.2	55.0	97.2	53.8	99.8	72.5	14.2%
	ILS	62.9	63.9	50.0	97.2	48.8	99.4	70.4	10.9%
	JGSA	**67.5**	62.3	55.6	98.1	52.0	99.8	72.5	14.2%
	DICE	**67.5**	**71.9**	**57.8**	97.2	**60.0**	**100.0**	**75.7**	**19.2%**
LP	DGA-DA	64.5	65.0	55.0	**97.2**	53.8	99.8	72.5	14.2%
	DICE$_{\text{lp}}$	**67.7**	**70.7**	**56.5**	**97.2**	**57.7**	**100.0**	**75.0**	**18.1%**
Non-1-NN	SVM	57.8	56.9	47.2	95.8	45.5	98.6	67.0	–
	SA*	59.4	57.7	47.2	95.1	46.5	99.0	67.5	0.7%
	CORAL	60.4	57.0	47.6	96.2	46.3	99.0	67.8	1.2%
	ATI [66]	**70.3**	68.7	55.3	95.0	56.9	98.7	74.2	10.7%
	DICE$_{\text{svm}}$	68.5	**72.5**	**58.1**	**97.2**	**60.3**	**100.0**	**76.1**	**13.6%**
End-to-end ‡	DDC [33]	64.4	61.8	52.1	95.0	52.2	98.5	70.6	0.7%
	DAN [34]	67.0	68.5	54.0	96.0	53.1	99.0	72.9	4.0%
	DANN [42]	72.3	73.0	53.4	96.4	51.2	99.2	74.3	† 4.1%
	DRCN [37]	66.8	68.7	56.0	96.4	54.9	99.0	73.6	5.0%
	RTN [12]	71.0	73.3	50.5	96.8	51.0	**99.6**	73.7	† 7.1%
	kNN-Ad [39]	**84.1**	**81.1**	**58.3**	96.4	**63.8**	99.2	**80.5**	† **15.6%**
	WDAN [69]	64.5	66.8	53.8	95.9	52.7	98.7	72.1	8.1%
	JAN [36]	71.8	74.9	**58.3**	96.6	55.0	99.5	76.0	8.4%
	ADDA [43]	-	75.1	-	**97.0**	-	**99.6**	-	-
	AutoDIAL [13]	73.6	75.5	58.1	96.6	59.4	99.5	77.1	† 10.0%

† We calculate the gain over baseline network from the origin papers. ‡ Some results are cited from [36] where the baseline AlexNet scores a 70.1% Avg. acc.

5.3 Results of Basic Domain-invariant Projection

In this section, we compare DICE and its extensions with corresponding peer methods for unsupervised adaptation.

Results on the Office31 Dataset Table 2 summarizes the comparison results of Office31. We compare DICE with several shallow methods including JGSA and ILS, DICE$_{\text{lp}}$ with DGA-DA that also utilizes LP, and DICE$_{\text{svm}}$ with several non-1-NN counterparts including CORAL and ATI. Moreover, we show ten state-of-the-art (SOTA) end-to-end AlexNet based DA methods for comparison. Apparently, DICE, DICE$_{\text{lp}}$ and DICE$_{\text{svm}}$ outperform their peer methods in terms of Avg., among which ATI is the SOTA shallow method. Compared with deep DA methods w.r.t. the

2. The code is available online at https://tinyurl.com/y7gu2wg8.
3. https://www.csie.ntu.edu.tw/~cjlin/liblinear/
4. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/

TABLE 3
Recognition accuracies (%) on the **Office-Caltech** dataset using the protocol [9], [10]. The best value is highlighted in bold and red.

data	1-NN	GFK-pls	SA	JDA	DIP-CC	OTGL	CDDA	ILS	JGSA	DICE	$JGSA_{lin}$	$JGSA_{rbf}$	$DICE_{lin}$	$DICE_{rbf}$	SVM	SA*	CORAL	$DICE_{svm}$	DGA-DA	$DICE_{lp}$
A→C	26.0	**43.6**	39.8	39.4	40.0	34.6	39.6	40.0	41.5	42.7	38.1	41.5	43.8	**44.1**	35.6	44.3	**45.1**	44.3	38.0	**44.1**
A→D	25.5	44.6	36.9	39.5	36.9	38.9	38.2	40.1	47.1	**49.7**	45.9	45.2	**48.4**	43.3	36.3	36.3	39.5	**52.2**	42.0	**49.0**
A→W	29.8	45.1	37.6	38.0	35.9	37.0	46.1	39.0	45.8	**52.2**	49.5	45.1	46.1	**52.9**	31.9	38.3	44.4	**53.2**	52.2	**52.9**
C→A	23.7	**51.6**	42.1	44.8	40.8	44.2	49.4	48.5	51.5	50.2	52.3	53.1	**56.7**	52.1	42.9	**54.8**	54.3	53.8	50.3	**53.7**
C→D	25.5	43.3	45.9	45.2	45.2	44.5	49.7	45.9	45.9	**51.0**	**48.4**	**48.4**	44.6	47.1	33.8	45.2	36.3	**51.6**	**52.9**	51.6
C→W	25.8	44.1	32.2	41.7	37.3	38.9	38.6	41.4	45.4	**48.1**	45.8	48.5	**51.9**	45.1	34.6	44.4	38.6	**54.9**	45.8	**53.9**
D→A	28.5	39.1	34.2	33.1	31.5	37.2	32.8	**41.2**	38.0	41.1	36.0	38.7	**47.7**	35.3	34.3	39.4	37.7	**42.6**	37.1	**41.2**
D→C	26.3	31.7	32.5	31.5	30.6	32.4	33.7	**34.6**	29.9	33.7	30.2	30.3	**37.7**	33.5	32.1	**34.3**	33.8	34.2	33.5	**34.5**
D→W	63.4	85.4	88.5	89.5	83.7	81.1	89.8	85.8	**91.9**	84.1	91.9	**93.2**	84.7	88.5	78.0	**85.1**	84.7	84.1	**91.2**	84.1
W→A	23.0	31.8	34.3	32.8	27.6	39.4	36.7	37.6	**39.9**	37.5	**41.0**	40.8	38.6	37.8	37.5	36.3	35.9	**39.5**	32.4	**33.1**
W→C	19.9	34.3	28.8	31.2	28.8	36.0	32.0	31.2	33.2	**37.8**	32.7	33.6	34.7	**35.8**	33.9	33.2	33.7	**38.6**	31.2	**37.8**
W→D	59.2	87.9	88.5	89.2	**91.7**	84.0	91.1	86.0	90.5	87.3	90.5	88.5	86.0	**94.3**	80.9	83.4	86.6	**87.3**	**91.7**	87.3
Avg.	31.4	48.5	45.1	46.3	44.2	45.7	48.1	47.6	50.0	**51.3**	50.2	50.6	**51.7**	50.8	42.6	47.9	47.6	**53.0**	49.8	**51.9**

SURF ⇑ and $DeCAF_6$ ⇓ features. (Two state-of-the-art (SOTA) shallow methods JDOT [65], ATI [66] and two SOTA end-to-end method RTN [12], AutoDIAL [13] are also compared.)

data	1-NN	GFK-pls	SA	JDA	DIP-CC	OTGL	CDDA	ILS	JGSA	DICE	$JGSA_{lin}$	$DICE_{lin}$	DGA-DA	$DICE_{lp}$	SVM	SA*	CORAL	$DICE_{svm}$	JDOT	ATI	RTN	AutoDIAL
A→C	83.7	82.1	80.8	85.0	78.9	85.5	85.8	77.9	84.9	**85.9**	85.0	**85.8**	86.7	**86.9**	85.0	85.0	83.6	**87.6**	85.2	86.5	**87.8**	87.4
A→D	80.3	82.2	86.0	86.6	80.9	85.0	86.6	79.0	88.5	**89.8**	85.4	**90.5**	89.2	**89.8**	87.9	87.3	84.7	91.1	87.9	**92.8**	**92.9**	-
A→W	74.6	73.2	76.3	83.1	69.8	83.1	81.4	82.0	81.0	**86.4**	**84.8**	79.7	86.4	**88.8**	79.0	76.6	74.2	88.1	84.8	**88.7**	**93.8**	-
C→A	90.0	92.1	89.4	91.0	89.8	92.2	91.6	87.4	91.4	**92.3**	**91.8**	91.3	92.3	**92.9**	91.4	92.2	91.2	93.4	91.5	**93.8**	93.2	**94.3**
C→D	86.6	92.4	90.4	87.9	84.7	87.3	91.7	91.1	**93.6**	**93.6**	**92.4**	**92.4**	**92.4**	91.1	89.8	88.5	87.9	**95.5**	89.8	89.6	**93.9**	90.1
C→W	78.6	84.4	80.0	82.4	72.2	84.2	88.8	77.6	86.8	**93.6**	**85.1**	84.1	89.8	**93.2**	80.0	82.0	80.7	**95.3**	88.8	93.6	**96.6**	96.3
D→A	85.7	88.4	87.1	91.0	85.3	92.3	92.1	88.7	92.0	**92.5**	92.3	**92.6**	92.4	**92.7**	87.1	85.6	83.8	92.5	88.1	**93.4**	**93.6**	-
D→C	79.2	80.3	81.4	85.1	75.5	84.1	86.3	79.5	86.2	**87.4**	85.8	**87.1**	86.5	**87.6**	78.8	76.9	71.6	**88.5**	84.3	85.9	83.4	**86.9**
D→W	99.7	99.0	98.3	**99.7**	98.6	96.3	99.0	98.6	**99.7**	99.0	98.6	**99.0**	**99.0**	**99.0**	98.6	96.9	97.6	**99.0**	96.6	98.9	**98.6**	-
W→A	77.1	84.3	83.7	**91.1**	72.4	90.6	90.4	86.4	90.7	90.7	91.4	**93.1**	90.7	**90.9**	75.7	83.6	72.1	91.1	90.7	**93.6**	**92.7**	-
W→C	74.8	76.5	79.3	85.3	70.3	81.5	**85.5**	79.1	85.0	85.3	**84.7**	82.9	85.6	**86.0**	72.0	74.3	67.4	**88.0**	82.6	86.3	84.8	**86.8**
W→D	**100**	**100**	98.7	**100**	99.4	96.3	**100**	99.4	**100**	**100**	**100**	**100**	**100**	**100**	99.4	99.4	**100**	**100**	98.1	**100**	**100**	-
Avg.	84.2	86.2	85.9	89.1	81.5	88.2	89.9	85.6	90.0	**91.4**	89.8	**89.9**	90.9	**91.6**	85.4	85.7	82.9	**92.5**	89.0	91.9	**92.6**	-

† CDDA and its variant DGA-DA [16] are carefully implemented by ourselves, the original averaged accuracies of them with SURF features are 48.2% and 49.0%, and 89.1% and 90.8% for $DeCAF_6$ features, respectively. Unless specified otherwise, all the results of CDDA and DGA-DA are reproduced by ourselves.

TABLE 4
Averaged recognition accuracies (%) on the **Office-Caltech** dataset with various features under the 'splitting' protocol [67], [68].

feature	SURF (800) [67], [68]														VGG-FC_6 (4,096) [11]						VGG-FC_7 (4,096) [11]					
data	1-NN	GFK-pls	SA	JDA	DIP-CC	CDDA	ILS	JGSA	DICE	SVM	SA*	CORAL	$DICE_{svm}$	DGA-DA	$DICE_{lp}$	1-NN	JDA	CDDA	ILS	JGSA	DICE	1-NN	JDA	CDDA	ILS	JGSA	DICE
A→C	22.8	**39.7**	34.5	34.0	34.9	38.0	38.0	34.1	38.8	35.6	39.8	**40.4**	40.1	39.1	**39.6**	78.9	80.9	82.1	79.1	76.4	**83.6**	77.5	78.0	80.7	78.5	80.4	**84.3**
A→D	22.0	38.2	32.3	38.8	32.5	36.2	37.5	36.6	**39.4**	34.2	36.2	37.3	**39.6**	37.3	**39.7**	61.1	70.6	68.2	**71.9**	69.7	66.0	57.6	67.6	63.2	**70.8**	70.0	65.3
A→W	25.1	40.5	32.1	39.9	34.5	39.1	38.6	37.6	**41.2**	32.3	37.8	38.8	**41.8**	40.8	**42.9**	70.1	81.0	78.1	**81.9**	75.0	76.6	70.6	76.8	77.1	80.0	**80.1**	77.7
C→A	19.2	36.4	31.9	37.2	32.6	40.3	41.0	31.3	**43.8**	34.9	39.7	39.6	**45.5**	41.2	**44.8**	80.6	87.3	86.5	86.1	89.4	**89.5**	80.7	87.1	85.9	86.8	87.3	**89.4**
C→D	19.5	37.8	32.7	36.4	33.9	37.6	**40.9**	34.1	40.0	33.8	37.0	38.9	**40.1**	38.8	**39.6**	59.3	69.9	66.1	70.3	**72.0**	69.9	56.6	66.0	63.8	66.9	**69.0**	67.0
C→W	17.1	32.6	27.1	32.4	29.4	33.6	35.6	36.0	**39.2**	30.0	32.7	33.3	**39.6**	35.5	**40.5**	66.8	80.4	77.1	**80.6**	78.8	79.8	64.4	77.5	75.3	81.3	**81.4**	80.4
D→A	26.9	37.8	32.2	34.7	34.1	37.6	39.6	**40.8**	**40.8**	34.3	36.8	37.3	**42.2**	38.7	**40.9**	64.0	77.9	82.6	79.0	81.8	**83.2**	61.2	71.6	76.9	76.9	76.1	**78.9**
D→C	24.6	32.6	30.7	29.4	31.8	32.8	**34.4**	26.8	33.3	31.3	33.3	**33.7**	33.3	33.4	**33.8**	61.2	73.8	76.1	67.6	73.2	**78.7**	59.3	65.3	71.3	67.9	71.0	**76.5**
D→W	52.8	80.1	79.9	**84.2**	78.2	83.7	79.4	75.0	81.3	74.3	79.0	80.8	**81.4**	**84.4**	81.7	93.9	94.4	93.7	93.6	92.4	**95.8**	91.3	90.2	92.1	90.9	93.9	**94.9**
W→A	20.7	36.7	31.9	35.8	32.6	36.8	37.1	32.5	**38.4**	35.5	35.8	37.0	**38.9**	37.8	**38.7**	71.2	87.8	86.5	85.8	**89.1**	88.8	71.2	**89.4**	83.8	85.0	88.2	89.2
W→C	17.1	30.3	26.5	28.7	28.2	32.0	31.5	24.5	**33.7**	31.0	31.1	31.6	**34.2**	32.5	**34.5**	67.3	80.1	80.1	76.8	76.3	**82.0**	66.8	76.9	79.2	75.0	79.6	**82.9**
W→D	42.0	70.5	71.8	75.7	72.3	**77.0**	69.4	64.4	75.7	68.2	70.8	72.3	**75.7**	**77.6**	76.1	91.3	**93.9**	92.8	87.5	89.8	88.1	88.6	89.1	90.4	84.9	**92.3**	87.5
Avg.	25.8	42.8	38.6	42.2	39.6	43.7	43.6	39.5	**45.5**	39.6	42.5	43.4	**46.0**	44.8	**46.1**	72.2	81.5	80.8	80.0	80.3	**81.8**	70.5	77.9	78.3	78.7	80.8	**81.2**

† The results of ILS differ those reported in [11] since 8 instead of 20 samples per class is utilized here for source domain C for fair comparisons.

averaged accuracy, $DICE_{svm}$ is only inferior to AutoDIAL and kNN-Ad, yet, it beats all of them but kNN-Ad in terms of the accuracy gain over baseline net. $DICE_{svm}$ also obtains the highest accuracies on two small-scale tasks D⇆W and acceptable performance for relatively large source A.

Results on the Office-Caltech Dataset As stated before, we carry out various experiments for comparisons on this classic dataset. First, we study the case: SURF and $DeCAF_6$ features under the 'full training' protocol in Table 3. Compared with 1-NN based methods, DICE obtains the best average accuracy while wining 5 out of 12 cross-domain pairs, and the second-best method JGSA merely wins on D→W and W→A. Generally, DICE achieves the 2nd or 3rd highest results on other 5 tasks except for W⇄D. Regarding the average accuracy, DICE also has a large advantage which improves 2.6% over JGSA. Additionally, we compare the kernelized extensions of DICE and JGSA with linear and RBF kernels, it is obvious to find that $DICE_{lin}$ and $DICE_{rbf}$ achieve the best accuracies in the majority of tasks, respectively. Carefully looking the average accuracy, DICE is always superior to JGSA for both kernels. On one hand, exploiting the SVM classifier, $DICE_{svm}$ outperforms SA* and CORAL with a much larger improvement. On the other hand, leveraging the label propitiation technique, $DICE_{lp}$ also beats CDDA's variant DGA-DA. Note that all these extensions achieve better performance than DICE.

Following [10], we also perform the comparison experiment with high-dimensional $DeCAF_6$ features, via which all the methods obtain higher results. Concerning the average accuracy, DICE easily defeats the remaining shallow methods including OTGL and CDDA and wins 9 out of 12 tasks, and JGSA is the second-best method. Besides, $DICE_{lin}$, $DICE_{lp}$ and $DICE_{svm}$ consistently outperforms their peer methods, including $JGSA_{lin}$, DGA-DA and SA*. Even compared with SOTA shallow approaches, JDOT and ATI, $DICE_{svm}$ still win 8 out of 12 tasks and improves the average accuracy by 3.8% and 0.5% respectively. RTN [12] and AutoDIAL [13] are two SOTA deep DA methods based on AlexNet, whereas, $DICE_{svm}$ still obtains competitive results with them. Particularly, $DICE_{svm}$ outperforms RTN in 6 out of 12 tasks and AutoDIAL in 4 out of 6 tasks.

Second, we also compare our methods with SOTA approaches (e.g., CORAL [27] and ILS [11]) under the 'splitting' protocol in Table 4. DICE is the best performing method in 7 out 12 tasks and outperforms other 1-NN based shallow methods in terms of the average accuracy. ILS and CDDA are second-best performing methods that are only inferior to DICE. However, the average accuracy of JGSA is 39.5%, which is much worse in this protocol. Besides, $DICE_{lp}$ also easily beats DGA-DA in 10 out of

TABLE 5
Recognition accuracies (%) on the **PIE** dataset with two preprocessing tools, l_2-normalization and z-score standardization († last 6 columns).

data	1-NN	GFK-pls	SA	JDA	DIP-CC	OTGL	CDDA	ILS	JGSA	DICE	SVM	SA*	CORAL	DICE_{svm}	DGA-DA	DICE_{lp}	1-NN†	GFK-pls†	JDA†	CDDA†	JGSA†	DICE†
P1→P2	26.1	29.4	26.8	73.1	29.9	59.4	76.3	37.8	62.2	**84.1**	30.9	32.8	31.8	**84.2**	76.4	**83.9**	37.8	42.2	76.1	77.5	46.8	**85.1**
P1→P3	26.6	28.3	28.2	69.3	32.8	58.7	72.3	35.2	60.0	**77.9**	33.9	34.5	31.9	**77.5**	72.5	**77.5**	46.4	53.7	74.1	77.0	48.3	**86.9**
P1→P4	30.7	34.6	30.9	90.1	36.7	-	92.1	36.6	80.6	**95.9**	41.4	43.5	41.8	**95.8**	92.1	**95.9**	62.5	69.8	92.8	90.3	81.0	**96.2**
P1→P5	16.7	21.8	19.6	55.1	12.7	48.4	60.7	21.8	45.1	**66.5**	23.8	22.5	19.9	**66.9**	60.8	**66.0**	36.3	43.9	70.8	67.8	44.1	**71.8**
P2→P1	24.5	30.0	26.4	73.8	25.8	61.9	77.0	40.4	68.2	**81.3**	31.8	27.7	26.6	**82.4**	77.0	**81.4**	42.1	43.2	**79.8**	77.3	65.1	76.8
P2→P3	46.6	45.5	48.0	**74.9**	53.4	64.4	77.5	31.8	64.9	74.0	41.0	37.3	35.0	**74.0**	77.5	**74.1**	55.2	54.0	80.2	**81.1**	54.0	79.1
P2→P4	54.1	57.1	54.3	83.8	50.1	-	87.1	45.9	77.6	**88.6**	62.2	58.5	59.7	**89.5**	87.1	**88.4**	66.3	69.1	90.4	88.5	82.8	**93.5**
P2→P5	26.5	27.7	28.2	61.5	29.5	52.7	64.3	25.5	52.3	**68.8**	28.8	27.1	25.9	**70.8**	63.6	**68.0**	41.1	42.6	68.3	70.4	44.8	**72.3**
P3→P1	21.4	26.9	23.2	69.0	22.7	57.9	**80.8**	29.1	62.9	78.8	32.3	29.1	25.1	**81.1**	**80.8**	78.0	41.4	51.6	78.3	**81.2**	61.9	80.1
P3→P2	41.0	41.4	44.3	74.5	36.3	64.7	72.2	31.9	60.3	**76.7**	39.7	37.0	36.5	**78.1**	72.2	**75.9**	54.0	52.0	81.2	**82.0**	55.9	77.7
P3→P4	46.5	51.3	46.2	82.1	45.8	-	84.7	44.3	71.0	**85.2**	61.9	54.8	54.0	**86.8**	84.7	**85.2**	66.0	72.1	92.3	91.7	84.5	**95.1**
P3→P5	26.2	31.7	28.9	60.4	20.2	52.8	64.3	18.0	51.2	**70.8**	37.7	30.5	26.0	**71.4**	64.5	**71.3**	45.8	50.9	70.5	**79.9**	53.6	78.1
P4→P1	33.0	42.8	36.3	90.2	31.4	-	**93.6**	48.1	84.4	93.3	57.7	52.4	48.3	**94.1**	**93.4**	93.3	64.5	72.6	95.1	89.7	83.6	**96.8**
P4→P2	62.7	64.5	63.8	91.9	67.5	-	93.2	50.1	83.5	**95.0**	69.2	70.0	69.7	**95.9**	93.2	**95.0**	72.8	75.8	94.7	94.8	78.0	**96.6**
P4→P3	73.2	73.3	73.2	90.1	76.8	-	92.2	63.2	80.8	**92.3**	69.7	72.7	72.7	**92.5**	92.2	**92.3**	78.2	80.9	92.4	92.1	76.2	**94.5**
P4→P5	37.2	44.7	38.1	69.4	36.5	-	74.0	40.6	65.9	**81.1**	48.7	48.6	48.5	**81.9**	74.0	**80.5**	52.9	61.1	80.8	85.1	61.4	**90.4**
P5→P1	18.5	31.4	23.4	59.6	14.2	45.7	68.1	25.7	53.5	**73.8**	29.4	34.5	32.0	**75.7**	67.7	**74.2**	30.4	45.3	64.1	67.3	50.0	**79.4**
P5→P2	24.2	28.2	25.5	67.5	29.3	51.3	65.1	21.4	57.5	**71.2**	33.1	30.9	30.4	**72.4**	65.4	**69.2**	34.0	38.9	74.2	**74.5**	52.7	71.4
P5→P3	28.3	34.4	28.6	69.5	31.7	52.6	70.5	31.7	54.3	**74.1**	40.6	31.9	32.6	**75.8**	71.6	**74.6**	41.1	47.7	75.3	79.3	58.7	**82.7**
P5→P4	31.2	40.0	31.2	74.3	26.3	-	79.7	36.0	62.3	**81.8**	51.5	45.1	44.5	**83.7**	79.7	**83.5**	46.6	59.3	81.7	80.3	78.4	**89.5**
Avg.	34.8	39.2	36.3	74.0	35.5		77.3	35.8	64.9	**80.6**	43.3	41.1	39.6	**81.5**	77.3	**80.4**	50.8	56.3	80.7	81.4	63.1	**84.7**

TABLE 6
Recognition accuracies (%) on the **Office-Home** dataset with ResNet50-P_5 features. (†ResNet152-P_5 features, ‡Results [59] based on a VGG-F network.)

data	1-NN	GFK-pls	SA	JDA	DIP-CC	CDDA	ILS	JGSA	DICE	DGA-DA	DICE_{lp}	CNN	SVM	SA*	CORAL	DICE_{svm}	1-NN†	JDA†	CDDA†	JGSA†	DICE†	DAN‡	DANN‡	DAH-e‡	DAH‡
Ar→Cl	35.9	35.8	36.1	40.5	35.5	40.8	37.4	40.8	**41.8**	40.8	**41.8**	36.0	36.5	36.7	36.3	**42.6**	39.4	42.8	42.6	41.9	**43.5**	30.7	**33.3**	29.2	31.6
Ar→Pr	54.4	54.4	54.6	58.9	54.3	57.7	55.6	58.2	**59.7**	57.7	**60.0**	53.7	54.7	54.7	54.1	**61.1**	58.4	62.9	62.3	63.1	**64.2**	42.2	**43.0**	35.7	40.8
Ar→Rw	64.9	65.0	65.0	67.5	64.9	66.3	65.3	67.5	**67.7**	66.3	**67.7**	64.6	64.9	65.3	65.3	**68.3**	68.2	70.0	69.2	70.1	**70.5**	54.1	**54.4**	48.3	51.7
Cl→Ar	39.5	39.2	39.4	40.8	39.5	41.3	39.9	40.8	**41.8**	41.3	**41.9**	39.2	40.6	39.9	39.2	**43.3**	45.1	45.7	46.3	46.5	**46.7**	32.8	32.3	33.8	**34.7**
Cl→Pr	48.4	48.4	47.9	51.9	48.0	51.7	47.6	52.0	**52.6**	51.7	**52.5**	48.9	48.1	48.3	47.9	**54.3**	49.1	54.5	52.9	54.1	**54.6**	47.6	49.1	48.2	**51.9**
Cl→Rw	51.4	51.8	51.9	55.2	51.9	53.9	52.2	54.6	**55.9**	53.9	**56.0**	51.7	52.5	51.4	51.5	**57.1**	56.1	59.0	57.8	59.0	**59.4**	49.8	49.8	47.5	**52.8**
Pr→Ar	41.8	42.3	42.2	45.1	41.9	46.1	42.6	45.3	**47.0**	46.1	**47.0**	41.2	42.0	41.4	41.5	**48.3**	47.4	49.7	50.5	50.4	**51.8**	29.1	**30.5**	29.9	29.9
Pr→Cl	32.4	32.5	32.3	33.3	32.1	**35.4**	32.9	33.5	34.4	**35.4**	34.3	32.9	32.6	33.0	32.7	**35.9**	39.9	41.6	43.0	42.0	**43.9**	34.1	38.1	38.8	**39.6**
Pr→Rw	64.1	64.1	63.8	67.2	63.7	66.0	64.0	66.4	**68.0**	66.0	**68.0**	63.2	64.4	64.4	64.1	**69.2**	67.7	70.8	69.6	70.5	**71.8**	56.7	56.8	55.6	**60.7**
Rw→Ar	58.1	58.1	58.3	58.8	58.2	59.1	57.5	58.5	**59.6**	59.1	**59.6**	57.9	57.7	57.8	57.7	**60.2**	62.4	63.4	63.1	62.3	**64.1**	43.6	44.7	41.2	**45.0**
Rw→Cl	39.5	39.5	40.0	44.2	39.2	45.3	42.2	43.8	**45.7**	45.3	**45.8**	39.5	40.1	40.0	39.5	**46.2**	43.4	47.0	47.9	46.2	**48.0**	38.3	42.7	45.0	**45.1**
Rw→Pr	69.6	69.6	69.6	72.4	69.6	71.6	70.6	72.4	**73.2**	71.6	**73.2**	68.9	69.8	69.5	69.3	**73.5**	71.3	73.5	73.8	73.1	**74.3**	62.7	**64.7**	59.1	62.5
Avg.	50.0	50.1	50.1	53.0	49.9	52.9	50.6	52.8	**54.0**	52.9	**54.0**	49.8	50.3	50.2	49.9	**55.0**	54.0	56.7	56.6	56.6	**57.7**	43.5	44.9	42.7	**45.5**

12 tasks. Once replacing the SURF features with VGG-FC features, the average accuracy of baseline 1-NN increases rapidly from 25.8% to nearly 71%. DICE is always the best performing method, while JDA and JGSA are merely inferior to DICE for different features respectively. Besides, the VGG-FC_6 features are more favorable for unsupervised domain adaptation methods.

Results on the PIE Dataset For the cross-domain facial image recognition task, we extensively compare DICE with several SOTA methods including OTGL [64], CDDA, and JGSA, with two data preprocessing tools in Table 5. l_2 normalization is a widely used tool especially for facial images, but inspired by SURF features for Office-Caltech, we also attempt to utilize z-score standardization for raw pixel features. We can clearly find that z-score standardization is much better than l_2 normalization for almost all methods. Regarding the basic baseline method 1-NN, the average accuracy increase from 34.8% to 50.8%. Compared with shallow methods including OTGL, DICE wins all 12 tasks with great scores, which leads the second-best method CDDA by 19.8%. Similarly, DICE_{svm} and DICE_{lp} also take significant advantages in comparison over SVM and DGA-DA. Surprisingly, once replaced with the z-score standardization, the improvement of JGSA is even negative, while JDA owns the largest improvement. For all cross-view face recognition tasks, DICE win 16 out of 20 tasks, to the best of our knowledge, it is the best performance on the PIE dataset.

Results on the Office-Home Dataset Office-Home [59] is a recently proposed benchmark dataset for cross-domain object recognition. Following [27], we fine-tune the pre-trained ResNet50 and ResNet152 models [32] on the ImageNet with the labeled source domain and extract the 5th pooling features for unsupervised domain adaptation.

TABLE 7
Recognition accuracies (%) on the **MNIST - USPS** and **COIL20** datasets.

data	1-NN	GFK	SA	JDA	DIP-CC	CDDA†	ILS	JGSA	DICE	DGA-DA†	DICE_{lp}	SVM	SA*	CORAL	DICE_{svm}
M→U	65.9	68.6	67.8	70.6	67.1	76.2	71.2	**80.4**	76.9	**82.3**	78.3	48.8	45.3	35.8	**79.7**
U→M	44.7	50.1	48.8	60.0	46.3	62.1	54.9	**68.2**	64.8	**70.8**	65.2	27.8	29.5	36.4	**59.8**
Avg.	55.3	59.3	58.3	65.3	56.7	69.1	63.0	**74.3**	70.9	**76.5**	71.8	38.3	37.4	36.1	**69.8**
C1→C2	83.6	86.4	86.8	94.7	84.6	91.5	86.9	95.4	**99.7**	99.6	**99.7**	77.6	83.2	82.1	**92.5**
C2→C1	82.8	85.0	85.0	93.5	84.0	93.9	86.9	93.9	**99.7**	**99.7**	**99.7**	74.6	82.5	81.8	**94.4**
Avg.	83.2	85.7	85.9	94.1	84.3	92.7	86.9	94.7	**99.7**	**99.7**	**99.7**	76.1	82.8	81.9	**93.5**

† The reproduced results are much lower than those reported in [16], hence we report the original results here.

Similarly, DICE_{lp} and DICE_{svm} also take significant advantages in comparison over DGA-DA and CORAL. Concerning the ResNet50-P_5 features, DICE achieves the best performance except Pr→Cl where DICE is only inferior to CDDA. While utilizing the ResNet152-P_5 features for several 1-NN methods, the results improve those with ResNet50-P_5 features by nearly 8%. Among them, DICE still performs the best while JDA, CDDA and JGSA are comparable to each other. The results indicate that our method can achieve much better performance regardless of the feature preprocessing. Besides, we also show the original results in [59] via a VGG-F network, our results can easily beat them in terms of the recognition accuracies.

Results on the MNIST-USPS & COIL20 Dataset To study the cross-domain task on digit images, we further carry out the experiments on the MNIST-USPS dataset [9] for comparisons. As can be seen from Table 7, JGSA obtains the best results among NN-based methods and DICE ranks the 2nd for both M→U and U→M and the average accuracy. DICE_{lp} enhances the recognition accuracies, it is still inferior to DGA-DA, however, the reported results of CDDA and DGA-DA are much higher than what we have reproduced. These two domains may perform differently, JGSA utilizes domain-specific projections which allows more flexibilities, making it better than DICE. DICE_{svm} is the best SVM-based method, which significantly outperforms the baseline methods. We also display some recent results of OTGL [64]

(U→M: 70.0, M→U: 57.9) and RTML [19] (U→M: 61.8), whereas DICE still beats them. Besides, DICE performs slightly better than deep WDAN [69] (U→M: 72.6, M→U: 65.4). Counter-intuitively, the 1-NN baseline method is even better than SVM on this dataset, and this may explain why DICE_{svm} is a bit lower than DICE.

The comparison results in Table 7 indicate that DICE achieves the highest accuracies in all the tasks and settings. Compared with RTML [19] (C1→C2: 91.2), DICE is still the better approach. Closer inspection shows that the performances are very close to 100% and SVM is again inferior to 1-NN, explaining the performance degradation after incorporating SVM to DICE.

5.3.1 Ablation Study & Parameter Analysis

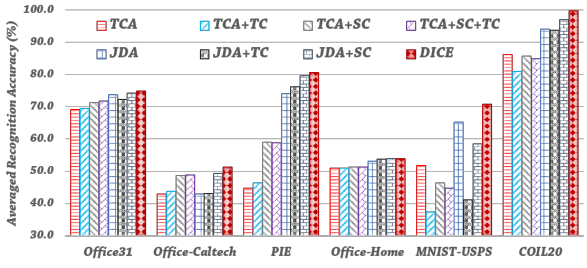


Fig. 3. Ablation study of our algorithm (SC/TC: source/target class clustering). [**Office-Caltech**: SURF & **PIE**: l_2-normalization & **Office-Home**: ResNet50-P_5]

To verify the effectiveness of the proposed algorithm, we perform an ablation study with its results shown in Fig. 3. It is important to clarify that TCA [29] is also a special case of DICE where $Q^{\text{all}} = Q^0$ in Eq. (14), i.e., only the first term in Eq. (7) is considered. Afterwards, we respectively incorporate source or target or both class clustering promoting terms to TCA and JDA, and compare them on six datasets. Regarding the former 4 datasets, both SC and TC can always enhance the performance of TCA and JDA, yet, they harm the performance for both methods on MNIST-USPS and TCA on COIL20 due to the possible accumulative errors in preliminary iterations. Roughly speaking, JDA performs better than TCA and DICE performs better than both, which indicates that all three terms are essential in our method.

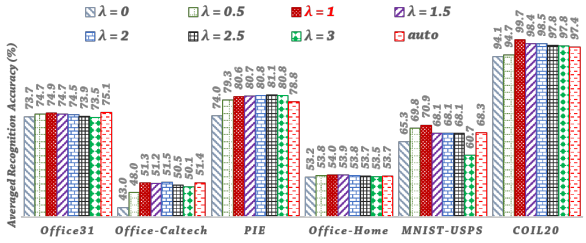


Fig. 4. Parameter study of our algorithm: averaged accuracy with parameter λ. [**Office-Caltech**: SURF & **PIE**: l_2-normalization & **Office-Home**: ResNet50-P_5]

In addition, we conduct experiments to investigate the sensitivity of our method to the weighting parameter λ. Besides the range of $[0, 0.5, \cdots, 3]$, we develop an automatic method dubbed 'auto', where λ is fixed to $(n_s + n_t)/n_s n_t$ without normalization in Step 7 of Algorithm 1. As can be seen from Fig. 4, as λ grows larger, the average accuracy of every dataset increases and then decreases gradually. As we expected, the optimal results are always achieved when λ is fixed to 1, even for Office-Caltech and PIE, '$\lambda = 1$' obtains competitive results with the highest accuracies (51.5% and 81.1%). Despite its inferiority to '$\lambda = 1$', the 'auto' method is still acceptable that is much higher than JDA ($\lambda = 0$). Generally, the class weighting term is quite effective and '$\lambda = 1$' is the optimal tradeoff where the inter-domain and intra-domain objectives are balanced. Besides, we discuss the convergence of our method (target accuracies and t-SNE visualizations) in Algorithm 1 in the Supplemental Material.

5.4 Results of Random Ensemble Methods

As stated in Section 4.2 and Algorithm 2, we develop several fusion methods below to investigate the effectiveness of the proposed 'sampling-and-fusion' strategy. **Con-1-NN** and **Con-SVM** are two typical FF methods which concatenate all projected features learned on different sub-domain pairs and trains an overall 1-NN or linear SVM classifiers respectively; **MV-1-NN** and **MV-SVM** are two representative MV methods which count the votes from each 1-NN or SVM classifier built on different sub-domain pairs and decide the most favorable candidate category; **WMV-1-NN** is one typical example of WMV methods which considers the continuous votes instead of discrete votes. We adopt the same one-hot encoding as MV, that only one candidate class owns non-zero score, but the score is calculated as the ratio $\sum_{i=1}^{\tau} \delta_{y_1}(y_i)/\tau$, where the neighborhood size τ is fixed at 10, and y_i is the label of the i-th nearest neighbor. To exclude undesired randomness caused by sampling, we run each method 10 times and report the average and standard deviation for each method.

Clearly, this proposed fusion strategy is quite favorable for distributed system which can alleviate the computation burden from large-scale and high-dimensional datasets. Yet, we are more interested in whether such strategy improves the cross-domain recognition accuracy. To investigate the effectiveness and sensitivity, in Fig. 5 we plot the averaged recognition accuracies for an object dataset Office-Caltech and a digit dataset MNIST-USPS while the sampling densities $\delta_s, \delta_t, \delta_f$ vary in the range of $[0.1, 0.2, \cdots, 0.9]$ respectively. Then we carry out this ensemble methods with merely feature sampling on several high-dimensional (i.e., d > 1,000) datasets with results shown in Table 8.

As can be seen in Table 8, almost all random ensemble methods consistently outperform DICE and DICE_{svm} in terms of the averaged recognition accuracies. Each class distributes uniformly on the COIL20 dataset and the recognition accuracy of the basic method DICE is rather high, which may explain why the sampling strategy does not work well. However, these relatively lower accuracies are still much higher than that of the second-best method JGSA (94.7%) in Table 7. Once observing the differences between these ensemble methods, we find that WMV-1-NN wins MV-1-NN and Conv-1-NN and MV-SVM beats Conv-SVM in a majority of cases. This observation indicates that majority voting is a better choice than concatenating features for feature sampling. Looking carefully at averaged accuracies for each task on the PIE dataset, the ensemble methods based on both NN and SVM perform better than their baseline models even though the dimensionality is relatively smaller than general deep representations. Moreover, compared with the averages, the deviations are somewhat small, which indicate the results obtained from random ensemble methods are accurate and stable.

Besides the feature sampling, we also study the instance sampling on the source and target domains for two benchmark datasets shown in Figs. 5(a)(b)(d)(e), respectively. Roughly looking these figures, WMV-1-NN always achieves the best performance for different sampling choices, which also outperforms MV-SVM and Conv-SVM in a majority of sampling densities. Additionally, MV-1-NN and MV-SVM consistently outperforms Conv-1-NN and Conv-SVM respectively, which again validate that majority voting is a better choice than concatenating features for fusing different projections. When $\delta_s \geq 0.3$ or $\delta_t \geq 0.4$ or $\delta_f \geq 0.5$, all the ensemble methods based on NN are superior to DICE on the Office-Caltech dataset. That maybe because 10 object classes on the Office-Caltech are imbalanced and the label distributions of source and target are also significantly different. While for the digit dataset, only for large sampling densities like 0.7, 0.8, the fusion methods are better than DICE. As we expected, δ_f is the most active sampling density since the original feature dimensionalities (800, 256) here are rather low.

TABLE 8
Averaged recognition accuracy ± standard deviation (%) on five 'high-dimensional' (d > 1,000) datasets ($\delta_f = 0.5, K = 10$).

dataset	DICE	Conv-1-NN	MV-1-NN	WMV-1-NN	DICE_{svm}	Conv-SVM	MV-SVM
Office31	**75.7**	75.4±0.2	75.0±0.2	75.2±0.2	**76.1**	75.5±0.2	75.3±0.3
Office-Caltech	91.4	91.4±0.1	91.8±0.1	**92.2**±0.2	**92.4**	92.3±0.3	92.4±0.2
PIE (l_2-norm.)	80.6	81.6±0.2	81.6±0.2	**82.0**±0.2	81.5	82.5±0.2	**82.8**±0.3
Office-Home	54.0	54.1±0.1	54.1±0.0	**54.4**±0.0	55.0	54.5±0.0	**55.1**±0.0
COIL20	**99.7**	99.2±0.2	99.3±0.1	99.3±0.1	93.5	**95.7**±0.4	93.6±0.2
P1→P2	**84.1**	84.0±1.0	83.5±1.5	83.5±1.1	**84.2**	83.0±1.1	83.6±1.3
P1→P3	77.9	**79.4**±1.8	79.2±1.7	79.3±1.8	77.5	79.0±1.7	**80.0**±2.0
P1→P4	95.9	96.2±0.2	**96.6**±0.3	96.5±0.3	95.8	**97.0**±0.1	96.6±0.3
P1→P5	66.5	68.8±0.9	**69.6**±0.7	69.2±0.6	66.9	68.0±1.1	**69.5**±0.7
P2→P1	81.3	82.2±0.7	81.8±0.7	**82.5**±0.6	82.4	**83.7**±0.5	83.6±0.7
P2→P3	74.0	76.6±0.4	76.2±0.4	**76.9**±0.5	74.0	74.1±0.5	**76.2**±0.5
P2→P4	88.6	89.1±0.6	89.4±0.5	**89.8**±0.5	89.5	90.5±0.6	**90.6**±0.4
P2→P5	68.8	68.9±1.0	69.5±1.5	**70.3**±1.3	70.8	68.5±1.3	**71.1**±1.2
P3→P1	78.8	78.7±0.7	78.7±0.9	**79.4**±1.0	**81.1**	79.3±0.8	80.6±0.9
P3→P2	76.7	77.8±0.7	77.3±1.0	**77.9**±1.1	78.1	**78.6**±0.9	78.6±1.1
P3→P4	85.2	87.3±0.6	87.4±0.7	**88.1**±0.6	86.8	**89.4**±0.7	89.4±0.6
P3→P5	70.8	70.9±0.8	71.5±0.6	**71.8**±0.6	71.4	71.7±0.9	**73.0**±0.8
P4→P1	93.3	**94.5**±0.3	94.2±0.5	93.9±0.5	94.1	**96.2**±0.4	95.0±0.5
P4→P2	95.0	94.7±0.4	95.0±0.2	**95.2**±0.2	95.9	**96.3**±0.2	96.1±0.2
P4→P3	92.3	92.0±0.2	91.9±0.5	**92.0**±0.5	**92.5**	92.3±0.3	92.0±0.3
P4→P5	81.1	81.4±0.4	81.3±0.5	**81.8**±0.4	81.9	**82.9**±0.4	82.6±0.6
P5→P1	73.8	75.6±1.1	76.2±1.2	**76.3**±1.2	75.7	**78.6**±0.7	78.1±1.1
P5→P2	71.2	**73.7**±1.4	72.8±1.5	73.6±1.3	72.4	**75.8**±1.0	75.3±1.4
P5→P3	74.1	76.5±0.4	76.6±0.8	**77.6**±0.6	75.8	77.3±0.8	**78.8**±0.7
P5→P4	81.8	83.4±0.8	83.1±0.7	**83.7**±0.7	83.7	**88.0**±0.7	85.6±0.8

5.4.1 Discussion of Hybrid Sampling

Both Fig. 5 and Table 8 describe the results of ensemble methods with single kind of sampling, that is to say, only one of sampling densities $[\delta_s, \delta_t, \delta_f]$ is less than 1.0. In this section, we are also interested in whether fusion with hybrid sampling works better than fusion with single sampling. For simplicity, we exploit the Office-Caltech dataset with SURF features and determine several relative large values for the sampling densities with fixed K 10, the comparison results of WMV-1-NN for each setting are shown in Table 9.

There exist two hybrid sampling settings, (i.e., DICE_{st} and DICE_{stf}) and three single sampling settings (i.e., DICE_{s}, DICE_{t} and DICE_{f}). First, we underline the results which are higher than its counterpart method DICE without sampling. Generally, methods with single sampling $(\delta_s, \delta_t, \delta_f)$ win DICE in 6, 9, and 11 out of 12 tasks in terms of the averaged accuracy, meanwhile, methods with hybrid sampling DICE_{st} and DICE_{stf} beat DICE in 9 and 10 sub-tasks, respectively. Among these fusion methods, DICE_{stf} is superior to both DICE_{st} and DICE_{f}, while DICE_{st} outperforms DICE_{t} but slightly inferior to DICE_{s}. It indicates that hybrid sampling show great potential to enhance the adaptation performance at a lower computation cost.

Re-checking the results, we find that DICE_{s} often performs the best among the former 6 tasks where source domains A, C are relatively large. Likewise, DICE_{t} achieves promising results for small-scale source domains D and W. Combining both source and target instance sampling together, DICE_{st} not only advances the averaged accuracy of DICE_{t} but also obtains a lower deviation value 0.20 for the overall dataset. This finding also works well on two medium-sized sub-tasks A⇄C. In addition to instance sampling, DICE_{stf} further incorporates the feature sampling and achieves the best performance except A→W, C→D. For such cases, source domain is much larger than target domain, sampling on the target instances severely destroy the distribution, making the learned projection sub-optimal.

TABLE 9
Averaged recognition accuracy ± standard deviation (%) on the **Office-Caltech** dataset with SURF features under the 'full-training' protocol for WMV-1-NN w.r.t. different sampling densities $[\delta_s, \delta_t, \delta_f]$.

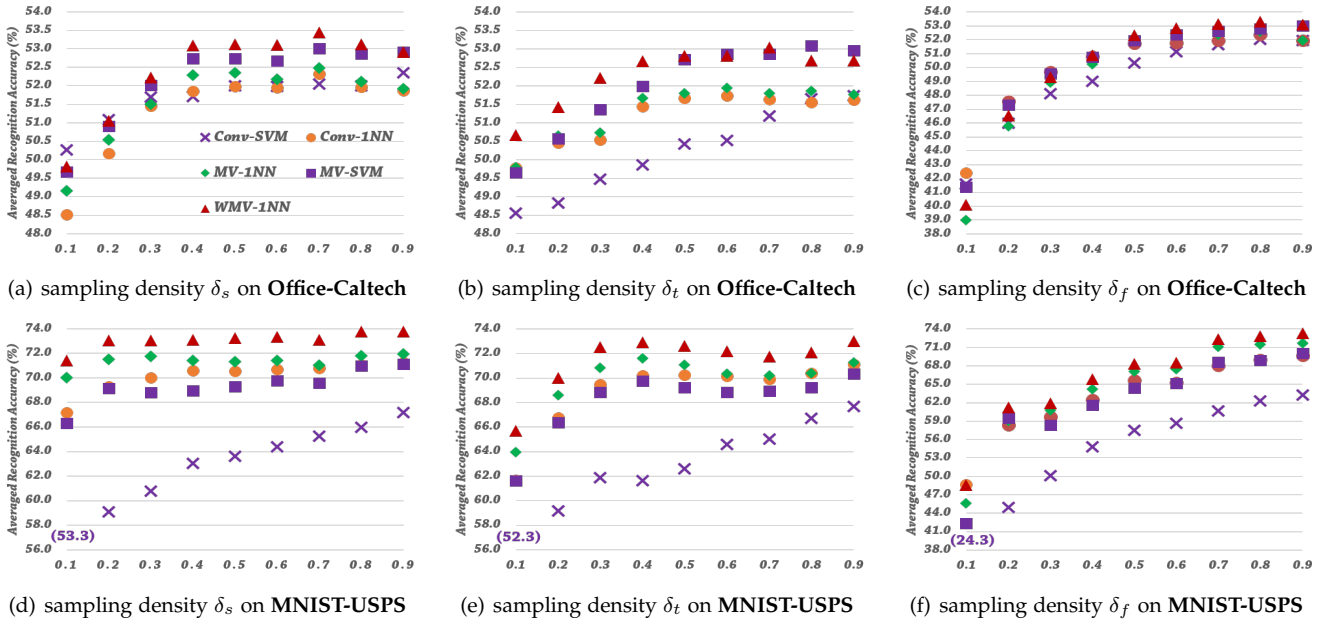
data	DICE	DICE_{s}	DICE_{t}	DICE_{st}	DICE_{f}	DICE_{stf}
$[\delta_s, \delta_t, \delta_f]$	–	[0.8, 1.0, 1.0]	[1.0, 0.6, 1.0]	[0.8, 0.6, 1.0]	[1.0, 1.0, 0.9]	[0.8, 0.6, 0.9]
A→C	42.65	45.00±0.63	46.23±0.19	**46.43**±0.53	45.02±0.47	46.16±0.50
A→D	49.68	**51.97**±1.99	50.19±1.52	50.06±1.72	49.68±1.91	**51.97**±1.07
A→W	52.20	52.00±1.28	49.49±1.83	49.90±1.18	**52.41**±0.88	49.08±1.25
C→A	50.21	**55.59**±0.85	53.55±0.80	54.55±0.65	54.66±0.72	54.82±0.55
C→D	50.96	**55.29**±2.36	53.25±0.97	54.01±2.68	54.27±1.05	53.50±2.85
C→W	48.14	**56.27**±0.90	51.73±0.57	54.24±0.79	54.17±0.73	54.24±0.83
D→A	41.13	43.55±0.50	43.32±0.22	**43.70**±1.54	43.63±1.43	42.73±0.73
D→C	33.66	35.32±0.59	35.64±0.80	35.87±0.32	35.98±0.34	**36.06**±0.52
D→W	84.07	82.51±1.28	**86.24**±0.78	85.15±0.88	84.20±0.78	85.63±1.26
W→A	37.47	35.89±0.95	40.56±0.72	41.32±0.29	37.49±1.08	**41.50**±1.18
W→C	37.85	37.47±1.59	36.54±1.20	35.55±0.71	**37.95**±0.96	35.90±0.76
W→D	87.26	86.62±0.78	87.01±0.35	86.11±1.23	87.52±0.73	**87.52**±0.57
Avg.	51.27	53.12±0.44	52.81±0.08	53.07±0.20	53.08±0.33	**53.26**±0.32

5.4.2 Robustness, Generalization and Large-scale Case

Additionally, we study several important issues including robustness to source label distribution, generalization ability of the learned projections and performance on a large-scale digit dataset SVHN→MNIST [39], [43]. Specifically, we measure the robustness under uniform/non-uniform source label distributions, while generalization ability represents the adaptation performance via learning projections for classes 'A,B', and predicting cross-domain classes 'C,D'. More details can be found in the Supplementary Material.

6 Conclusion

In this work, we have analyzed the domain-irrelevant class-clustering objective, and derived a novel objective function to learn domain-invariant projection for unsupervised domain adaptation. The optimal projection and pseudo target labels are alternately optimized, and in each iteration the projection is computed in closed-form via solving a generalized eigenvalue problem, while the pseudo labels are estimated via discriminative classifier trained on projected source domain. To increase the discriminative ability of final classifier, we first introduce the 'sampling-and-fusion' strategy into the domain adaptation task, where multiple independent projections are optimized on coupled domain subsets. Actually, this ensemble method can be naturally parallelized and be flexible for large-scale and high-dimensional datasets. Extensive experimental results demonstrate that our methods converge fast in terms of recognition accuracy and achieve performances superior or comparable to state-of-the-art approaches.



(a) sampling density δ_s on **Office-Caltech** (b) sampling density δ_t on **Office-Caltech** (c) sampling density δ_f on **Office-Caltech**

(d) sampling density δ_s on **MNIST-USPS** (e) sampling density δ_t on **MNIST-USPS** (f) sampling density δ_f on **MNIST-USPS**

Fig. 5. Averaged recognition accuracy (%) on the **Office-Caltech** dataset with SURF features using the 'full training' protocol (a-c) and **MNIST-USPS** dataset (d-f) w.r.t. each sampling density δ_s, δ_t and δ_f while the other two densities are kept fixed at 1.

We believe that our methods are readily extended to semi-supervised and multi-source unsupervised domain adaptation tasks. This would be useful for some scenarios where partial target data are labeled or multiple labeled source domains exist. In this sense, label propagation can be introduced to handle labeled target data, and multi-kernel learning can be easily leveraged in multi-source fusion. Besides, multiple domain-invariant projections are learned independently, which does not consider the explicit correlations between each projection. One interesting direction is to collaboratively learn multiple projections, which is expected to achieve more promising results. We intend to investigate these problems in future work.

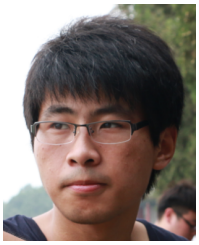
ACKNOWLEDGMENTS

The authors would like to greatly thank the associate editor and the reviewers for their valuable comments and advices, and Dr. Dangwei Li and Ying Mao for helpful discussions. This work was funded by State Key Development Program (Grant No. 2016YFB1001001), and National Natural Science Foundation of China (Grant Nos. 61622310, 61473289). Zhenan Sun and Tieniu Tan are the corresponding authors.

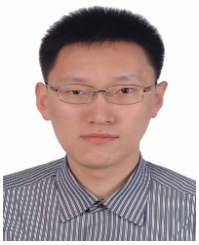
REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
[2] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. ICCV*, 2011.
[3] W.-S. Chu, F. De la Torre, and J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," in *Proc. CVPR*, 2013.
[4] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2288–2302, 2014.
[5] M. Baktashmotlagh, M. Harandi, and M. Salzmann, "Distribution-matching embedding for visual domain adaptation," *Journal of Machine Learning Research*, vol. 17, no. 108, pp. 1–30, 2016.
[6] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. EMNLP*, 2006.
[7] H. Daumé III, "Frustratingly easy domain adaptation," *arXiv preprint:0907.1815*, 2009.
[8] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, "Transfer learning with graph co-regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1805–1818, 2014.
[9] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. ICCV*, 2013.
[10] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. CVPR*, 2017.
[11] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *Proc. CVPR*, 2017.
[12] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. NIPS*, 2016.
[13] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "Autodial: Automatic domain alignment layers," in *Proc. ICCV*, 2017.
[14] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. NIPS*, 2007.
[15] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proc. ICCV*, 2013.
[16] L. Luo, X. Wang, S. Hu, C. Wang, Y. Tang, and L. Chen, "Close yet distinctive domain adaptation," *arXiv preprint:1704.04235*, 2017.
[17] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation." in *Proc. ICML*, 2013.
[18] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*, ser. Springer series in statistics. Springer, 2009.
[19] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, 2017.
[20] L. Luo, L. Chen, S. Hu, Y. Lu, and X. Wang, "Discriminative and geometry aware unsupervised domain adaptation," *arXiv preprint:1712.10042*, 2017.
[21] M. Dudík, R. E. Schapire, and S. J. Phillips, "Correcting sample selection bias in maximum entropy density estimation," in *Proc. NIPS*, 2005.
[22] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, "Covariate shift by kernel mean matching," *Dataset Shift in Machine Learning*, pp. 131–160, 2009.
[23] 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, 2014.
[24] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. CVPR*, 2012.

[25] 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, 2012.
[26] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. ICCV*, 2013.
[27] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016.
[28] P. Koniusz, Y. Tas, and F. Porikli, "Domain adaptation by mixture of alignments of second-or higher-order scatter tensors," in *Proc. CVPR*, 2017.
[29] 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, 2011.
[30] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, 2017.
[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. ECCV*, 2016.
[33] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint:1412.3474*, 2014.
[34] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015.
[35] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. ECCV Workshops*, 2016.
[36] M. Long, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. ICML*, 2017.
[37] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. ECCV*, 2016.
[38] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. NIPS*, 2016.
[39] O. Sener, H. O. Song, A. Saxena, and S. Savarese, "Learning transferrable representations for unsupervised domain adaptation," in *Proc. NIPS*, 2016.
[40] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint:1409.7495*, 2014.
[41] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. ICCV*, 2015.
[42] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.
[43] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. CVPR*, 2017.
[44] D. Lopez-Paz and M. Oquab, "Revisiting classifier two-sample tests," *arXiv preprint:1610.06545*, 2016.
[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014.
[46] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. NIPS*, 2016.
[47] B. Chidlovskii, S. Clinchant, and G. Csurka, "Domain adaptation in the absence of source domain data," in *Proc. ACM SIGKDD*, 2016.
[48] Y.-H. Hubert Tsai, Y.-R. Yeh, and Y.-C. Frank Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proc. CVPR*, 2016.
[49] W.-Y. Chen, T.-M. H. Hsu, Y.-H. H. Tsai, Y.-C. F. Wang, and M.-S. Chen, "Transfer neural trees for heterogeneous domain adaptation," in *Proc. ECCV*, 2016.
[50] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, 2012.
[51] L. Niu, W. Li, and D. Xu, "Exploiting privileged informtion from web dt for action nd event recognition," *Int'l J. Computer Vision.*, vol. 118, no. 2, pp. 130–150, 2016.
[52] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.
[53] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proc. ICML*, 2012.
[54] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain generalization and adaptation using low rank exemplar svms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1114–1127, 2018.
[55] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
[56] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, 2004.
[57] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," 2010.
[58] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.
[59] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. CVPR*, 2017.
[60] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, 2003.
[61] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.
[62] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[63] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," 1996.
[64] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, 2017.
[65] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Proc. NIPS*, 2017.
[66] P. Panareda Busto and J. Gall, "Open set domain adaptation," in *Proc. ICCV*, 2017.
[67] J. Hoffman, E. Rodner, T. Darrell, J. Donahue, and K. Saenko, "Efficient learning of domain-invariant image representations," in *Proc. ICLR*, 2013.
[68] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko, "Asymmetric and category invariant feature transformations for domain adaptation," *Int'l J. Computer Vision.*, vol. 109, no. 1-2, pp. 28–41, 2014.
[69] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. CVPR*, 2017.



Jian Liang received the B.E. degree in Electronic Information and Technology from Xi'an Jiaotong University in July 2013. Now he is pursuing a Ph.D. degree in Pattern Recognition and Intelligent Systems in Center for Research on Intelligent Perception and Computing and National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). His research interests focus on machine learning, pattern recognition, and computer vision.



Ran He received the B.E. degree in Computer Science from Dalian University of Technology, the M.S. degree in Computer Science from Dalian University of Technology, and Ph.D. degree in Pattern Recognition and Intelligent Systems from CASIA in 2001, 2004 and 2009, respectively. Since September 2010, Dr. He has joined NLPR where he is currently a full Professor. He serves as an associate editor of Neurocomputing (Elsevier), and serves on the program committee of several conferences. His research interests focus on information theoretic learning, pattern recognition, and computer vision.

Zhenan Sun received the BE degree in industrial automation from Dalian University of Technology, China, the MS degree in system engineering from Huazhong University of Science and Technology, China, and the PhD degree in pattern recognition and intelligent systems from CASIA, in 1999, 2002, and 2006, respectively. He is currently a professor in the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, CASIA, China. His research interests include biometrics, pattern recognition, and computer vision. He is a fellow of the IAPR.

Tieniu Tan received his B.Sc. degree in electronic engineering from Xi'an Jiaotong University, China, in 1984, and his M.Sc. and Ph.D. degrees in electronic engineering from Imperial College London, U.K., in 1986 and 1989, respectively. He is currently a professor in the Center for Research on Intelligent Perception and Computing, NLPR, CASIA, China. He has published more than 450 research papers in refereed international journals and conferences in the areas of image processing, computer vision and pattern recognition, and has authored or edited 11 books. His research interests include biometrics, image and video understanding, information hiding, and information forensics. He is a fellow of the CAS, the TWAS, the BAS, the IEEE, the IAPR, the UK Royal Academy of Engineering, and the Past President of IEEE Biometrics Council.