arXiv:2003.11723v2 [cs.LG] 26 Jun 2021

Learning transferable and discriminative features for unsupervised domain adaptation

Yuntao Du [a] and Ruiting Zhang [a] and Xiaowen Zhang [a] and Yirong Yao [a] and
Hengyang Lu [a,b] and Chongjun Wang [a,*]

[a] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China*
E-mails: duyuntao@smail.nju.edu.cn, rayting1111@gmail.com, zhangxw@smail.nju.edu.cn,
yaotx@smail.nju.edu.cn, chjwang@nju.edu.cn
[b] *School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China*
E-mail: luhengyang@jiangnan.edu.cn

Abstract. Although achieving remarkable progress, it is very difficult to induce a supervised classifier without any labeled data. Unsupervised domain adaptation is able to overcome this challenge by transferring knowledge from a labeled source domain to an unlabeled target domain. Transferability and discriminability are two key criteria for characterizing the superiority of feature representations to enable successful domain adaptation. In this paper, a novel method called *learning TransFerable and Discriminative Features for unsupervised domain adaptation* (TFDF) is proposed to optimize these two objectives simultaneously. On the one hand, distribution alignment is performed to reduce domain discrepancy and learn more transferable representations. Instead of adopting *Maximum Mean Discrepancy* (MMD) which only captures the first-order statistical information to measure distribution discrepancy, we adopt a recently proposed statistic called *Maximum Mean and Covariance Discrepancy* (MMCD), which can not only capture the first-order statistical information but also capture the second-order statistical information in the reproducing kernel Hilbert space (RKHS). On the other hand, we propose to explore both local discriminative information via manifold regularization and global discriminative information via minimizing the proposed *class confusion* objective to learn more discriminative features, respectively. We integrate these two objectives into the *Structural Risk Minimization* (RSM) framework and learn a domain-invariant classifier. Comprehensive experiments are conducted on five real-world datasets and the results verify the effectiveness of the proposed method.

Keywords: Transfer learning, Unsupervised domain adaptation, Discriminative feature

1. Introduction

Supervised learning has achieved remarkable progress in many fields with the help of a large number of labeled training samples [1]. However, when there are few and even no labeled samples, it is difficult to, if not impossible, induce a supervised classifier. Rather, there is a need for versatile algorithms that reduce the need for large labeled datasets across multiple domains. Unsupervised domain adaptation address this need by transferring knowledge from a different but related domain (source domain) with labeled samples to a target domain with unlabeled samples to improve the performance of the target domain [2]. For example, an object classification model trained on manually annotated images may not generalize well to new images obtained under substantial variations in pose, occlusion, or light. Domain adaptation aims to enable knowledge transfer from the labeled source domain to the unlabeled target domain by exploring domain-invariant features that bridge different domains [3].

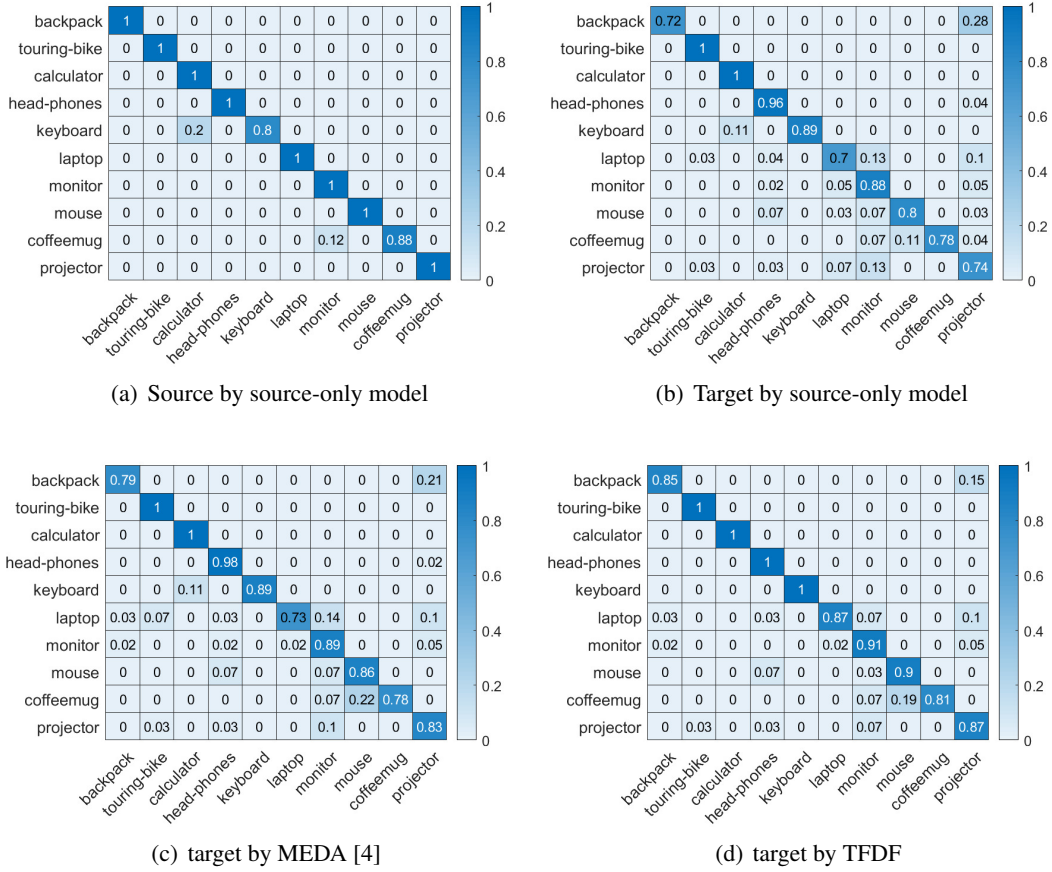*Corresponding author. E-mail: chjwang@nju.edu.cn.

(a) Source by source-only model

(b) Target by source-only model

(c) target by MEDA [4]

(d) target by TFDF

Fig. 1. The error matrix of different methods on tasks D → W of Offcie-Caltech dataset. Source-only model means the model trained with only labled data in the source domain. (a) Source-only model tested in the source domain (b) Source-only model tested in the target domain, (c) Model trained with MEDA (d) Model trained with TFDF. The results show that TFDF can effevtively avoid class confusion in the target domain and performs much better than other methods.

Transferability and discriminability are two key criteria that characterize the superiority of feature representations to enable domain adaptation [3–7]. The transferability indicates the ability of feature representations to bridge the discrepancy across domains, and we can effectively transfer a learning model from the source domain to the target domain via the transferable feature representations [3, 4, 7]. Discriminability refers to the ability to separate different categories easily by a supervised classifier trained on the feature representations, and the model can achieve better classification performance via the discriminative feature representations [5, 6].

Since the source samples and target samples are drawn from different distributions, it is important to reduce the distribution discrepancy across domains to learn transferable features. The mostly used shallow domain adaptation approaches include instance reweighting [8, 9] and distribution alignment [3, 4, 7]. The former assumes that a certain portion of the samples in the source domain can be reused for learning in the target domain and the samples from the source domain can be reweighted according to the relevance to the target domain. While the latter assumes that there exists a common space where the distributions of two domains are similar and focus on finding a feature transformation that projects
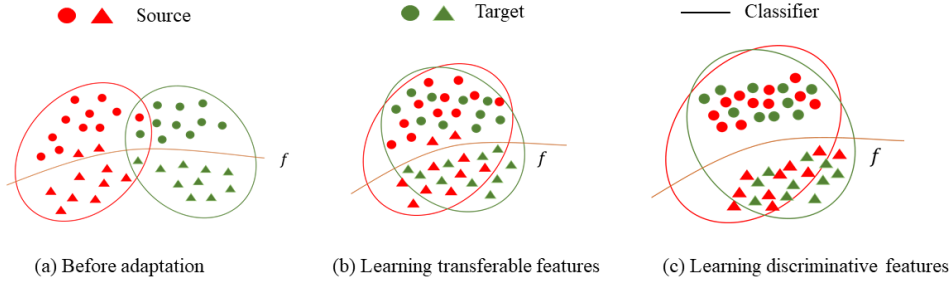
Fig. 2. An overview of the proposed method. (a) Before adaptation, the source classifier can not perform well in the target domain. (b)-(c) After learning transferable and discriminative features, the domain discrepancy can be reduced and the samples can be classified correctly.

features of two domains into another common subspace with less distribution discrepancy [3, 4, 7]. *Maximum Mean Discrepancy* (MMD) [10] based methods are popular methods for distribution alignment, where the MMD distance is used to evaluate the distribution discrepancy across domains.

While achieving remarkable progress, the experiments in [5] indicate that previous domain adaptation methods tend to enhance the transferability at the expense of deteriorating the discriminability. Thus, some methods, including geometrical based methods [11] and manifold regularization based methods [6, 12], also aim to improve the discriminability of the feature representations. Geometrical based methods consider the geometric divergence between both domains or the variance information in the target domain. Manifold regularization based methods are inspired by *manifold assumption* [13], which can make the predicted label of a certain sample consistent with its neighbor samples.

However, there are two issues with the existing methods. (1) To learn transferable features, MMD distance is a widely used statistic to measure the distribution discrepancy by kernel mean embedding of distributions. However, MMD distance only measures the first-order statistic of different distributions in reproducing kernel Hilbert space (RKHS). Some recent experiments have revealed that the second-order statistic (such as CORAL [14]) is also important to capture useful information for evaluating distribution discrepancy, which is ignored by many methods. (2) To learn discriminative features, previous methods mainly focus on local discriminative information (i.e., sample-level discriminative information), but ignore the global discriminative information (i.e., class-level discriminative information). For example, the classifier trained in the source domain may confuse to distinguish the correct class from a similar class [15], such as backpack and video-projector. As shown in Fig 1(a)-(b), the probability that a source-only model (only trained with labeled source data) misclassifies backpacks as video-projectors in the target domain is over 28%. This phenomenon is named *class confusion* and it reminds us that the global discriminative information should also be considered.

To overcome these issues, in this paper, we propose a novel method called *learning TransFerable and Discriminative Features for unsupervised domain adaptation* (TFDF), which learns a domain-invariant classifier under the principle of *Structural Risk Minimization* (SRM) to solve the above two issues simultaneously. An overview of the proposed method is shown in Fig 2. For the first issue, we adopt the recently proposed statistic called *Maximum Mean and Covariance Discrepancy* (MMCD) [16] to measure and decrease the distribution discrepancy across domains. MMCD is comprised of MMD and *Maximum Covariance Discrepancy* (MCD). MCD evaluates the Hilbert–Schmidt norm of the difference between covariance operators and can measure the second-order statistic in the RKHS. Therefore, MMCD can consider the first-order and the second-order statistics simultaneously in the RKHS and can

capture more distribution information than MMD. For the second issue, we aim to learn more discriminative features at both local and global levels. At the local level, we use the manifold regularization to further exploit the similar geometrical property of the nearest points. At the global level, instead of focusing on the feature space, we concentrate on the label space. We consider the confusion relationship between different classes which is revealed by the inner product of the classifier predictions between different classes (shown in Fig 1). The goal is that no examples are ambiguously classified into two classes at the same time. Thus we force the inner product of the same class close to 1 while the different classes close to 0, which encourages the samples in the same class to be more compact while the samples in the different classes to be more dispersed. Thus, TFDF can extract discriminative features.

To sum up, besides minimizing the empirical error in the source domain, TFDF also concentrates on minimizing the distribution discrepancy across domains to learn transferable features and exploring both global and local discriminative information to learn discriminative features. However, TFDF is a non-convex problem that is difficult to be solved directly, so we firstly propose a variant of TFDF named TFDF-V, which is a convex optimization problem that is easy to be solved with a closed-form solution. Then, taking the solution of TFDF-V as the initial value of TFDF, we use the Adam algorithm [17] (a variant of stochastic gradient descent) to solve the TFDF optimization problem. Comprehensive experiments on five different real-world cross-domain visual recognition datasets are conducted, and the results verify the effectiveness of the proposed algorithm.

2. Related Work

2.1. *Transferability in domain adaptation*

2.1.1. *Shallow domain adaptation*

Shallow domain adaptation methods include instance reweighting and distribution alignment. Instance reweighting based methods assume that the data from the source domain can be reused in the target domain by reweighting samples. Tradaboost [8] is the most representative method which is inspired by Adaboost [18]. The strategy of adjusting the weights of the source and target data is just the opposite, where the source data more conducive to the target data have greater weight in the source domain. LDML [19] also evaluates each sample, and takes full advantage of the pivotal samples, and filters out outliers. DMM [9] learns a transfer support vector machine by extracting invariant feature representations and estimating unbiased instance weights, to jointly minimize the cross-domain distribution discrepancy. However, the performance by instance reweighting is not satisfying.

Distribution alignment based methods focus on finding a feature transformation that projects features of two domains into another common subspace with less distribution discrepancy. The distribution discrepancy across domains includes marginal distribution discrepancy and conditional distribution discrepancy. TCA [3] tries to align marginal distribution across domains, which learns a domain-invariant representation during feature mapping. Based on TCA, JDA [7] tries to align both marginal distribution and conditional distribution simultaneously. Moreover, BDA [20] proposes a balance factor to leverage the importance of different distributions. MEDA [4] can dynamically evaluate the balance factor and has achieved promising performance. The above methods are all based on MMD, which only captures the first-order statistical information across domains. CORAL [14] explores the second-order statistic covariance of the target distribution. Many previous methods only adopt the first-order statitic information while ingore the second-order statistic information. Our method adopts MMCD [16] to evaluate the distribution discrepancy across domains and can capture more useful information for domain adaptation.

2.1.2. Deep domain adaptation

Most deep domain adaptation methods are based on statistical discrepancy minimization. DDC [21] embeds a domain adaptation layer into the Alexnet [22] and minimizes *Maximum Mean Discrepancy* (MMD) distance between features of this layer. DAN [23] minimizes the feature discrepancy between the last three layers of Alexnet [22] and the multiple-kernel MMD is used to measure the discrepancy. Other measures are also adopted such as Kullback-Leibler (KL) divergence, Correlation Alignment (CORAL) [24] which measures the second-order statistical information and Central Moment Discrepancy (CMD) [25] which measures the high-order statistical information. These methods can utilize the deep neural network to extract more transferable features and also have achieved remarkable performance.

Recently, Inspired by the generative adversarial network [26], adversarial learning is widely used in domain adaptation. DANN [27] adopts a domain discriminator to distinguish the source domain from the target domain, while the feature extractor is trained to learn domain-invariant features to confuse the discriminator. ADDA [28] designs a symmetrical structure where two feature extractors are adopted. Different from DANN, MCD [29] proposes a method to minimize the $H\Delta H$-distance across domains in an adversarial way.

2.2. Discriminability in domain adaptation

Learning transferable features may harm the discriminability of the features. Therefore, learning discriminative features is another objective for domain adaptation methods. Inspired by Linear discriminant analysis (LDA) [30], some methods take the geometrical information into consideration. For instance, the goal of JGSA [31] is to minimize the geometrical divergence across domains to enhance the discriminability in shallow domain adaptation. JJDA [32] extends this idea to deep domain adaptation and considers the instance-level discriminative information. Besides, LPJT [12] considers the manifold regularization via fisher criterion. ARTL [6] and MEDA also use the manifold regularization via local samples. These methods mainly focus on local discriminative information while the global discriminative information is ignored. TFDF can learn both local and global discriminative information, thus making the features more discriminative.

3. Method

3.1. Problem Definition

In this paper, we focus on unsupervised domain adaptation. There are a source domain $\mathcal{D}_s = \{(x_s^1, y_s^1), ..., (x_s^{n_s}, y_s^{n_s})\}$ of n_s labeled source samples where $x_s^i \in \mathcal{X}_s, y_s^i \in \mathcal{Y}_s$, and a target domain $\mathcal{D}_t = \{x_t^1, ..., x_t^{n_t}\}$ of n_t unlabled target samples where $x_t^i \in \mathcal{X}_t$. We assume the feature space and label space are the same, i.e., $\mathcal{X}_s = \mathcal{X}_t = \mathbb{R}^d$ and $\mathcal{Y}_s = \mathcal{Y}_t = \{1, 2, ..., C\}$, while these distributions across domains are different. Especially, we assume the marginal distribution and conditional distribution are different across domains, i.e., $P_s(x_s) \neq P_t(x_t)$ and $Q_s(y_s|x_s) \neq Q_t(y_t|x_t)$. Our goal is to learn a classifier $f : x_t \rightarrow y_t$ to predict $y_t \in \mathcal{Y}_t$ for the target domain $\mathcal{D}_t$ using samples from both domains.

3.2. Overall Objective

Transferability and discriminability are two key criteria that characterize the superiority of feature representations to enable domain adaptation [3–7]. Thus, TFDF aims to learn a domain-invariant classifier

f based on the principle of *Structural Risk Minimization* (SRM) to learn transferable and discriminative features for the distribution adaptation across domains. As mentioned before, we have four complementary objective functions as follows:

(1) Minimizing the source empirical error of the labeled data in the source domain.
(2) Minimizing the distribution discrepancy across domains to learn transferable features.
(3) Minimizing the manifold regularization to learn local discriminative features.
(4) Minimizing the proposed class confusion loss to learn global discriminative features.

The learning framework of TFDF is then formulated as:

$$f = \arg\min_{f \in \mathcal{H}_\mathcal{K}} R(f, \mathcal{D}_s) + \eta ||f||_K^2 + \lambda D_f(\mathcal{D}_s, \mathcal{D}_t) + \rho M_f(\mathcal{D}_s, \mathcal{D}_t) + \xi C_f(\mathcal{D}_s, \mathcal{D}_t) \tag{1}$$

where K is the kernel function induced by ϕ such that $\langle \phi(x_i), \phi(x_j) \rangle = K(x_i, x_j)$ and $\phi : \mathcal{X} \to \mathcal{H}$ is the feature mapping function that projects the original feature vector to a Hilbert space $\mathcal{H}$. $R(f, \mathcal{D}_s)$ is the empirical error in the source domain, $||f||_K^2$ is the squared norm of f. The term $D_f(\mathcal{D}_s, \mathcal{D}_t)$ represents the distribution discrepancy across domains, $M_f(\mathcal{D}_s, \mathcal{D}_t)$ is a Laplacian regularization and $C_f(\mathcal{D}_s, \mathcal{D}_t)$ is the class confusion loss. η, λ, ρ and ξ are the corresponding regularization parameters.

In the next subsections, we introduce each objective separately and give the learning method finally.

3.3. Source Error Minimization

The first objective of TFDF is to learn an adaptive classifier that can classify source samples correctly. To begin with, we can induce a standard classifier f on the labeled source samples. According to the structural risk minimization principle [33], we minimize the source empirical error as:

$$R(f, \mathcal{D}_s) + \eta ||f||_K^2 = \sum_{i=1}^{n_s} l(f(x_s^i), y_s^i) + \eta ||f||_K^2 \tag{2}$$

where $||f||_K^2$ is the squared norm of f in $\mathcal{H}_\mathcal{K}$ and $l(, \cdot,)$ is the loss function for classification. In TFDF, the squared loss $l(x_i, y_i) = (y_i - f(x_i))^2$ is used. According to the *Representer Theorem* [34], the classifier in optimization problem (1) can be represented as

$$f(x) = \sum_{i=1}^{n_s + n_t} \beta_i K(x_i, x) \tag{3}$$

and the equation (2) can be represented as:

$$\sum_{i=1}^{n_s} l(f(x_s^i), y_s^i) + \eta ||f||_K^2 = \sum_{i=1}^{n_s + n_t} A_{ii}(y_i - f(x_i))^2 + \eta ||f||_K^2 = ||(Y - \beta^T K)A||_F^2 + \eta \text{tr}(\beta^T K \beta) \tag{4}$$

where $A \in \mathbb{R}^{(n_s + n_t) \times (n_s + n_t)}$ is a diagonal label indicator matrix with $A_{ii} = 1$ if $x_i \in \mathcal{D}_s$, and $A_{ii} = 0$ otherwise. $Y \in \mathbb{R}^{C \times (n_s + n_t)}$ is the label matrix with $Y_{ij} = 1$ if x_j belongs to class i, and $Y_{ij} = 0$ otherwise. $K \in \mathbb{R}^{(n_s + n_t)(n_s + n_t)}$ is kernel matrix, and $\beta = (\beta_1, ..., \beta_{n_s + n_t}) \in \mathbb{R}^{(n_s + n_t) \times C}$ are the the parameters of the classifier.

3.4. Distribution Alignment

The distribution discrepancy across domains will result in performance degradation when directly applying the classifier trained in the source domain to the target domain. Thus, TFDF aims to learn transferable features to reduce the distribution discrepancy, which includes marginal distribution discrepancy and conditional distribution discrepancy.

Maximum Mean Discrepancy (MMD) is a widely used statistic to measure distribution distance, which compares different distributions $p(x)$ and $q(x)$ based on the distances between the sample means of two distributions in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, namely

$$MMD_{\mathcal{H}}^2[\mathcal{H}, p, q] = ||E_p\phi(x) - E_q\phi(x)||_{\mathcal{H}}^2 \tag{5}$$

However, MMD only measures the first-order statistic of different distributions. Some experiments have revealed that the second-order statistic (such as CORAL [14]) is also necessary to capture useful information for evaluating distribution discrepancy, which is ignored by existing methods. Recently, a new distribution metric termed *Maximum Mean and Covariance Discrepancy* (MMCD) is proposed in [16]. MMCD considers both the first-order and the second-order statistical information in the RKHS, which is defined as,

$$MMCD[p, q, \mathcal{H}] = (||E[p] - E[q]||_{\mathcal{H}}^2 + \gamma||C[p] - C[q]||_{\mathcal{HS}}^2)^{\frac{1}{2}} \tag{6}$$

where $C[p] = E_{x\sim p}[\phi(x) \otimes \phi(x)] - E_{x\sim p}[\phi(x)] \otimes E_{x\sim p}[\phi(x)]$, $||\cdot||_{\mathcal{HS}}$ denotes the *Hilbert-Schmidt* norm of the vectors in $\mathcal{HS}(\mathcal{H})$. The empirical estimator of the squared MMCD with classifier f can be given by [16],

$$\widehat{MMCD}^2[p, q, \mathcal{H}] = \text{tr}(\beta^T K M K^T \beta) + ||\beta^T K Z K^T \beta||_F^2 \tag{7}$$

where

$$M_{ij} = \begin{cases} \frac{1}{n_s^2}, & x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t^2}, & x_i, x_j \in \mathcal{D}_t \\ -\frac{1}{n_s n_t}, & \text{otherwise} \end{cases}, \quad Z_{ij} = \begin{cases} \frac{1}{n_s} - \frac{1}{n_s^2}, & i = j, x_i \in \mathcal{D}_s \\ -\frac{1}{n_s^2}, & i \neq j, x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t^2} - \frac{1}{n_t}, & i = j, x_i \in \mathcal{D}_t \\ \frac{1}{n_t^2}, & i \neq j, x_i, x_j \in \mathcal{D}_t \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Based on MMCD, the distribution discrepancy across domains can be written as

$$D_f(\mathcal{D}_s, \mathcal{D}_t) = (1 - \mu)D_{md}(\mathcal{D}_s, \mathcal{D}_t) + \mu D_{cd}(\mathcal{D}_s, \mathcal{D}_t) \tag{9}$$

where $D_{md}(\mathcal{D}_s, \mathcal{D}_t)$ and $D_{cd}(\mathcal{D}_s, \mathcal{D}_t)$ denote the marginal distribution discrepancy and conditional distribution discrepancy, respectively. μ is a balance factor between these two discrepancy. The distance of marginal distribution discrepancy is defined as the empirical distances across domains and the distance

of conditional probability distributions is defined as the sum of the empirical distances over the class labels between the sub-domains of a same label in the source and target domain,

$$
\begin{aligned}
D_{md}(\mathcal{D}_s, \mathcal{D}_t) &= \widehat{MMCD}^2[\mathcal{D}_s, \mathcal{D}_t, \mathcal{H}] = \text{tr}(\beta^T K M_0 K^T \beta) + ||\beta^T K Z_0 K^T \beta||_F^2 \\
D_{cd}(\mathcal{D}_s, \mathcal{D}_t) &= \sum_{c=1}^{C} \widehat{MMCD}^2[\mathcal{D}_{s,c}, \mathcal{D}_{t,c}, \mathcal{H}] = \sum_{c=1}^{C} (\text{tr}(\beta^T K M_c K^T \beta) + ||\beta^T K Z_c K^T \beta||_F^2)
\end{aligned}
\tag{10}
$$

where

$$
(M_0)_{ij} = \begin{cases} \frac{1}{n_s^2}, & x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t^2}, & x_i, x_j \in \mathcal{D}_t \ , \\ -\frac{1}{n_s n_t}, & otherwise \end{cases}
\qquad
(M_c)_{ij} = \begin{cases} \frac{1}{n_{s,c}^2}, & x_i, x_j \in \mathcal{D}_{s,c} \\ \frac{1}{n_{t,c}^2}, & x_i, x_j \in \mathcal{D}_{t,c} \\ -\frac{1}{n_{s,c} n_{t,c}}, & \begin{cases} x_i \in \mathcal{D}_{s,c}, x_j \in \mathcal{D}_{t,c} \\ x_j \in \mathcal{D}_{s,c}, x_i \in \mathcal{D}_{t,c} \end{cases} \\ 0, & otherwise \end{cases}
\tag{11}
$$

$$
(Z_0)_{ij} = \begin{cases} \frac{1}{n_s} - \frac{1}{n_s^2}, & i = j, x_i \in \mathcal{D}_s \\ -\frac{1}{n_s^2}, & i \neq j, x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t^2} - \frac{1}{n_t}, & i = j, x_i \in \mathcal{D}_t \\ \frac{1}{n_t^2} & i \neq j, x_i, x_j \in \mathcal{D}_t \\ 0 & otherwise \end{cases}
, (Z_c)_{ij} = \begin{cases} \frac{1}{n_{s,c}} - \frac{1}{n_{s,c}^2}, & i = j, x_i \in \mathcal{D}_{s,c} \\ -\frac{1}{n_{s,c}^2}, & i \neq j, x_i, x_j \in \mathcal{D}_{s,c} \\ \frac{1}{n_{t,c}^2} - \frac{1}{n_{t,c}}, & i = j, x_i \in \mathcal{D}_{t,c} \\ \frac{1}{n_{t,c}^2} & i \neq j, x_i, x_j \in \mathcal{D}_{t,c} \\ 0 & otherwise \end{cases}
\tag{12}
$$

where $\mathcal{D}_{s,c}(\mathcal{D}_{t,c}) = \{x_i | x_i \in \mathcal{D}_s(\mathcal{D}_t), y_i(\hat{y}_i) = c\}$, $y_i(\hat{y}_i)$ is the label (pseudo label) of the sample $x_s^i(x_t^i)$ and $n_{s,c}(n_{t,c}) = |\mathcal{D}_{s,c}(\mathcal{D}_{t,c})|$. As the term $||\beta^T K Z_c K^T \beta||_F^2$ is nonconvex, we can approximate the convex upper bound of the term in (10) by using the following theorem:

Theorem 1. *Given the constraint that $\beta^T K H K^T \beta = I$, the following inequality holds*

$$
||\beta^T K Z_c K^T \beta||_F^2 \leqslant \sigma k ||\beta^T K Z_c K^T||_F^2
\tag{13}
$$

where k is the feature dimensionality and $\sigma = ||(KHK)^{-\frac{1}{2}}||^2$.

where $H = I - \frac{1}{(n_s + n_t)}\mathbf{1}\mathbf{1}^T$ is the centering matrix and the proof is shown in appendix. Note that the constrain that $\beta^T K H K^T \beta = I$ is widely used in previous method [3, 7, 16] to avoid yielding a trivial

solution. According to theorem 1, we relax the objective (9) as

$$D_f(\mathcal{D}_s, \mathcal{D}_t) = (1 - \mu) \left(\text{tr}(\beta^T K M_0 K^T \beta) + ||\beta^T K Z_0 K^T||_F^2 \right) +$$

$$\mu \left(\sum_{c=1}^{C} (\text{tr}(\beta^T K M_c K^T \beta) + ||\beta^T K Z_c K^T||_F^2) \right)$$

$$= (1 - \mu) \left(\text{tr}(\beta^T K M_0 K^T \beta) + \text{tr}(\beta^T K Z_0 K^T K Z_0 K^T \beta) \right) + \tag{14}$$

$$\mu \left(\sum_{c=1}^{C} (\text{tr}(\beta^T K M_c K^T \beta) + \text{tr}(\beta^T K Z_c K^T K Z_c K^T \beta)) \right)$$

3.5. Local Discriminative Information

In domain adaptation, we except to learn discriminative features for better classification and adaptation. In this work, local discriminative information refers to the sample-level discriminative information. Manifold regularization is a widely used method to extract local discriminative features. According to the *manifold assumption* [13], if two points are close in the intrinsic geometry, then the corresponding labels are similar. Under this assumption, the *manifold regularization* is computed as

$$M_f(\mathcal{D}_s, \mathcal{D}_t) = \sum_{i,j=1}^{n_s+n_t} (f(x_i) - f(x_j))^2 W_{ij} = \sum_{i,j=1}^{n_s+n_t} f(x_i) L f(x_j) = \text{tr}(\beta^T K L K \beta) \tag{15}$$

where $W_{i,j}$ is the graph affinity matrix between x_i and x_j. $L = I - G^{-\frac{1}{2}} W G^{-\frac{1}{2}}$ is the graph Laplacian matrix, G is a diagonal matrix with $G_{ii} = \sum_{j=1}^{n+m} W_{ij}$, W is defined as

$$W_{ij} = \begin{cases} cos(x_i, x_j), & if \quad x_i \in N_p(x_j) \vee x_j \in N_p(x_i) \\ 0, & otherwise \end{cases} \tag{16}$$

where $N_p(x_i)$ is the set of p-nearest neighbors of x_i.

3.6. Global Discriminative Information

Maximizing manifold regularization can only learn local discriminative information from the nearest neighbors while the global discriminative information (i.e., class-level discriminative information) is ignored. As shown in Fig 1(a)-(b), there exists *class confusion* in domain adaptation methods, which means that the classifier trained in the source domain may confuse to distinguish the correct class from a similar class, such as backpack and video-projector. In order to solve this problem, instead of focusing on the feature space, we concentrate on the label space, where the prediction outputs are able to reveal the class relationships. we use the prediction outputs of the samples in both domains to minimize the class confusion. The prediction of the classifier f on both domains is defined as

$$\hat{F} = f(X) = \beta^T K \tag{17}$$

where $\hat{F} \in \mathbb{R}^{C*(n_s+n_t)}$, we recall that F_{ij} reveals the relationship between the j-th example and the i-th class. We define the pairwise class confusion between two classes i and j as

$$B_{ij} = \hat{F}_{i.}^T \cdot \hat{F}_{j.} \tag{18}$$

Note that $\hat{F}_{i.}$ denotes the probabilities that the examples in the target domain come from the i-th class. The class confusion is defined as the inner product between $\hat{F}_{i.}$ and $\hat{F}_{j.}$. So it measures the possibilities of classifying the examples in the target domain into the i-th and the j-th classes simultaneously.

Recall that B_{ij} well measures the confusion between class i and j. As we need to minimize the cross-class confusion, so the ideal situation is that no examples are ambiguously classified into two classes at the same time. In this case, the diagonal elements of B which represent the inner of same classes should be 1 while the off-diagonal elements which represent the inner of different classes should be 0 (as shown in Fig 1(a)). Therefore, our goal is to force B to approach the identity matrix. Then, we can define the class confusion objective as

$$\begin{aligned} C_f(\mathcal{D}_s, \mathcal{D}_t) &= ||B - I||_F^2 = ||\hat{F}\hat{F}^T - I||_F^2 = ||\beta^t K (\beta^t K)^T - I||_F^2 \\ &= ||\beta^T KK^T \beta - I||_F^2 \\ &= \text{tr}(\beta^T KK^T \beta \beta^T KK^T \beta - 2\beta^T KK^T \beta + I) \end{aligned} \tag{19}$$

Note that previous methods only learn discriminative features at the instance level (local level). However, by enforcing different classes to seperate from each other, the proposed method can learn discriminative features at the class level (global level).

3.7. Optimization Algorithm

By substituting equation (4), (14), (15), (19) into equation (1), we can get the optimization problem as follows,

$$\begin{aligned} \beta = \arg\min_{\beta} J(\beta) &= ||(Y - \beta^T K)A||_F^2 + \eta \text{tr}(\beta^T K\beta) + \lambda \text{tr}(\beta^T KVK\beta) + \\ &\rho \text{tr}(\beta^T KLK\beta) + \xi ||\beta^T KK^T \beta - I||_F^2 \\ s.t. \quad &\beta^T KHK\beta = I, \end{aligned} \tag{20}$$

where $V = (1-\mu)(M_0 + Z_0 KKZ_0) + \mu \sum_{c=1}^{C}(M_c + Z_c KKZ_c)$.

Equation (20) is an optimization problem with constraints, which is difficult to be solved directly. We relax the problem as an unconstrained optimization problem, namely

$$\begin{aligned} \beta = \arg\min_{\beta} J(\beta) &= ||(Y - \beta^T K)A||_F^2 + \eta \text{tr}(\beta^T K\beta) + \lambda \text{tr}(\beta^T KVK\beta) + \\ &\rho \text{tr}(\beta^T KLK\beta) + \xi ||\beta^T KK^T \beta - I||_F^2 + \delta \text{tr}(\beta^T KHK\beta - I), \end{aligned} \tag{21}$$

Since the fifth term of equation (21) is a non-convex fourth-order term, the optimization problem doesn't have the closed-form solution. Therefore, we adopt the adaptive moment estimation (**Adam**)

Algorithm 1 TFDF-V

Input: Input data $X = [X_s, X_t]$, source labels Y^s, iterations number T, parameters λ, ρ, η, ξ and neighbor p.

Output: Initial value β_{init}.

1: Train a base classifier using $\mathcal{D}_s$ and then apply prediction on $\mathcal{D}_t$ to get its pseudo labels $\hat{Y}_t$.
2: Construct kernel matrix K, graph Laplacian matrix L by equation (16).
3: **for** $t = 1, 2, ..., T$ **do**
4:     Calculate the balance factor μ using equation (24) and construct MMCD matrix V by equation (11),(12) and (20).
5:     Calculate β_{init} by solving equation (23) and obtain adaptive classifier f by equation (3),
6:     Update the pseudo labels of $\mathcal{D}_t : \hat{Y}^t = f(X_t)$.
7: **end for**

Algorithm 2 : TFDF

Input: Input data $X = [X_s, X_t]$, source labels Y^s, iterations number T, learning rate α, the parameters $\lambda, \rho, \eta, \xi, \delta$ and neighbor p.
    Initialization: $\theta_1 = 0.9, \theta_2 = 0.999, \epsilon = 10^{-8}$,

Output: Adaptive classifier $f : X \rightarrow Y$.

1: Train a base classifier using $\mathcal{D}_s$ and then apply prediction on $\mathcal{D}_t$ to get its pseudo labels $\hat{Y}_t$.
2: Construct kernel matrix K, graph Laplacian matrix L by equation (16).
3: Calculate $\beta_0 = \beta_{init}$ by solving equation (23) for **TFDF-V**.
4: **for** $t = 1, 2, ..., T$ **do**
5:     Calculate the balance factor μ using equation (24) and construct MMCD matrix V by equation (11),(12) and (20).
6:     Get gradients g_t by equation (22).
7:     $m_t \leftarrow \theta_1 m_{t-1} + (1 - \theta_1)g_t, v_t \leftarrow \theta_2 v_{t-1} + (1 - \theta_2)g_t^2, \widehat{m}_t \leftarrow \frac{m_t}{1-\theta_1^t}, \widehat{v}_t \leftarrow \frac{v_t}{1-\theta_2^t}$.
8:     $\beta_t = \beta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$.
9:     Obtain adaptive classifier f by equation (3).
10:     Update the pseudo labels of $\mathcal{D}_t : \hat{Y}^t = f(X_t)$.
11: **end for**

---

algorithm [17] which is a variant of stochastic gradient descent (**SGD**) to solve β iteratively. Take the derivative of β and we will get

$$
\begin{aligned}
g_t = \frac{\partial J(\beta)}{\partial \beta} = & -2KAY^T + 2KAK\beta + 2\eta K\beta + 2\lambda KVK\beta \\
& + 2\rho KLK\beta + 2\delta KHK\beta + 2\xi KK\beta\beta^T KK\beta - 2\xi KK\beta
\end{aligned}
\tag{22}
$$

We experimentally found that it is important to set a proper initial value. Thus, we propose a variant of optimization problem (21) which is named as **TFDF-V**. We let $\xi = 0$ for optimization problem (21), and by setting the derivative of objective function to 0, we can get

$$
\beta_{init} = ((A + \lambda V + \rho L + \delta H)K + \eta I)^{-1}AY^T
\tag{23}
$$

Note that **TFDF-V** can be solved by a closed solution. We firstly run the **TFDF-V** algorithm to get β_{init}. Then β_{init} is set as the initial value of **TFDF** algorithm. The detailed pseudo codes of **TFDF-V** and **TFDF** are described in algorithm 1 and algorithm 2, respectively.

4. Experiments and evaluations

In this section, we evaluate the performance of TFDF by extensive experiments on five widely-used common datasets. Codes will be available online upon publication.

4.1. Data Preparation

We adopt five public image datasets: Office+Caltech, MNIST+USPS, and COIL, which are popular for domain adaptation methods and have been widely used in previous works. Note that there are no noise and no missing values in these three datasets.

The **Office-Caltech** dataset [35] consists of images from 10 overlapping object classes between Office31 and Caltech-256. Specifically, we have four domains, **C** (*Caltech-256*), **A** (*Amazon*), **W** (*Webcam*), and **D** (*DSLR*). By randomly selecting two different domains as the source domain and target domain respectively, we construct $3 \times 4 = 12$ cross-domain object tasks, e.g. **C → A**, **C → W**,..., **D → W**. Both 800 SURF [7] and 4,096 DeCaf6 [36] features are used for these datasets.

USPS (U) and **MNIST** (M) are standard digit recognition datasets containing handwritten digits from 0-9. USPS consists of 7291 training images and 2007 test images of size 16 × 16. MNIST consists of 60000 training images and 10000 test images of size 28 × 28. We construct two tasks: U → M and M → U. 256 SURF features are used for these datasets.

COIL20 contains 20 objects with 1440 images. When the object rotates on the turntable, the object is photographed from different angles every 5 degrees, so each object has 72 images. Each image is 32×32 pixels with 256 gray levels per pixel. Two subsets COIL1 and COIL2 are partitioned from the dataset in [7]. We construct one dataset COIL1 vs COIL2 by selecting all 720 images in COIL1 to form the source data, and all 720 images in COIL2 to form the target data. We construct two tasks: COIL1 → COIL2 and COIL2 → COIL1.

4.2. Baselines

We compare the performance of TFDF with traditional machine learning approaches, several state-of-the-art traditional approaches, and deep domain adaptation approaches:

- traditional machine learning approaches: 1-Nearest Neighbor (**1NN**), Support Vector Machine (**SVM**) and Principal Component Analysis (**PCA**),
- traditional domain adaptation approaches: Transfer Component Analysis (**TCA**) [3], Geodesic Flow Kernel (**GFK**) [35], Joint Distribution Alignment (**JDA**) [7], Transfer Joint Matching (**TJM**) [37], Adaptation Regularization (**ARTL**) [6], CORrelation ALignment (**CORAL**) [14], Scatter Component Analysis (**SCA**) [38], Joint Geometrical and Statistical Alignment (**JGSA**) [11], Distribution Matching Machine (**DMM**) [9], MMCD based Domain Adaptation(**McDA**) [16], Manifold Embedded Distribution Alignment (**MEDA**) [4], and Confidence-Aware Pseudo Label Selection (**CAPLS**) [39].
- deep domain adaptation approaches: **AlexNet** [22], Deep Domain Confusion (**DDC**) [21], Deep Adaptation Network (**DAN**) [23], Deep CORAL (**DCORAL**) [24], and a compact DNN (**DNN**) [40].

4.3. Experimental Setup

For fair comparison and following [3, 35], 1NN, SVM and PCA are trained on the labeled source data, and tested on the unlabeled target data; Other traditional domain adaptation methods (e.g. TCA,JDA) are performed on both the source data and the target data and tested to classify the unlabeled target data. All the baselines except MEDA are performed in original feature space. While MEDA [4] and TFDF firstly perform *manifold feature learning* to project the original feature to a new feature space with $z = g(x) = \sqrt{G}x$. G can be computed efficiently by singular value decomposition [35]. RBF kernel is used in the experiment. We adopt the method in MEDA [4] to estimate the balance factor μ, namely,

$$\mu = \frac{d_M}{d_M + \sum_{c=1}^{C} d_c} \tag{24}$$

where $d_A(\mathcal{D}_s, \mathcal{D}_t) = 2(1 - 2\epsilon(h))$ is A-distance [41], which denotes the error of a linear classifier discriminating the two domains $\mathcal{D}_s$ and $\mathcal{D}_t$. We compute the marginal distribution discrepancy as $d_M = d_A(\mathcal{D}_s, \mathcal{D}_t)$ and the conditional distribution discrepancy as $\sum_{c=1}^{C} d_c = \sum_{c=1}^{C} d_A(\mathcal{D}_{s,c}, \mathcal{D}_{t,c})$. Deep methods can be used to the original images and the results of deep domain adaptation methods are directly reported from their original papers wherever available.

Under our experimental setup, it is impossible to tune the optimal parameters using cross validation, since labeled and unlabeled samples are from different distributions. Thus following previous methods [7], we evaluate all methods by empirically searching the parameter space for the optimal parameter settings, and report the best results of each method. We set number of nearest numbers by searching $p \in \{5, 10, 15, 20, 30\}$ and we set adaptation regularization parameter $\lambda, \eta, \rho, \delta$ by searching $\lambda, \eta, \rho, \delta, \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$.

In the comparative study of TFDF, we set 1) $\eta = 0.1, \lambda = 10.0, \rho = 0.1, \delta = 0.01, p = 10$ for COIL dataset, 2) $\eta = 0.1, \lambda = 10.0, \rho = 1.0, \delta = 0.01, p = 10$ for digital and Office-Caltech dataset. Additionally, we set $T = 10$ for TFDF-V and $T = 100$, $\alpha = 0.0005$ for TFDF. The experiments on parameter sensitivity in later experiments (Section 4.6) indicate that TFDF stays robust with a wide range of parameter choices. We use classification accuracy on the test data as the evaluation metric, which is widely used in literature [7]:

$$accuracy = \frac{|x : x \in \mathcal{D}_t \wedge \hat{y}(x) = y(x)|}{|x : x \in \mathcal{D}_t|}. \tag{25}$$

$y(x)$ and $\hat{y}(x)$ are the ground truth and predicted labels for the target domain samples, respectively.

4.4. Experimental Results and Analysis

The results on five real-world cross-domain (object, digit and object) datasets are shown in Table 1, 2 and 3. From these results, we can draw several observations:

Firstly, TFDF achieves the best performance in most tasks (7/12 tasks) on the Office-Caltech dataset (SURF features). The average accuracy of TFDF on the Office-Caltech dataset (SURF features) is 54.1%, while the best baseline is MEDA with 52.7%. Compared with MEDA, the average performance is improved by 1.4%. The observations on the COIL and digital datasets are the same and the average performance improvement is 2.7% on five datasets. Since these results are obtained from a large number

Table 1

Accuracy (%) on Office-Caltech datasets using SURF features.

Task	1NN	SVM	PCA	TCA	GFK	JDA	TJM	CORAL	SCA	ARTL	McDA	MEDA	TFDF
C → A	23.7	53.1	39.5	45.6	46.0	43.1	46.8	52.1	45.6	44.1	43.5	56.5	**58.0**
C → W	25.8	41.7	34.6	39.3	37.0	39.3	39.0	46.4	40.0	31.5	44.4	**53.9**	52.9
C → D	25.5	47.8	44.6	45.9	40.8	49.0	44.6	45.9	47.1	39.5	50.1	50.3	**59.2**
A → C	26.0	41.7	39.0	42.0	40.7	40.9	39.5	45.1	39.7	36.1	41.0	43.9	44.4
A → W	29.8	31.9	35.9	40.0	37.0	38.0	42.0	44.4	34.9	33.6	44.4	**53.2**	51.2
A → D	25.5	44.6	33.8	35.7	40.1	42.0	45.2	39.5	39.5	36.9	42.7	45.9	**46.5**
W → C	19.9	28.8	28.2	31.5	24.8	33.0	30.2	33.7	31.1	29.7	**35.3**	34.0	33.9
W → A	23.0	27.6	29.1	30.5	27.6	29.8	30.0	36.0	30.0	38.3	37.4	42.7	**43.1**
W → D	59.2	78.3	89.2	91.1	85.4	**92.4**	89.2	86.6	87.3	87.9	89.17	88.5	90.5
D → C	26.3	26.4	29.7	33.0	29.3	31.2	31.4	33.8	30.7	30.5	34.8	34.9	**37.2**
D → A	28.5	26.2	33.2	32.8	28.7	33.4	32.8	37.7	31.6	34.9	36.7	41.2	**42.1**
D → W	63.4	52.5	86.1	87.5	80.3	89.2	85.4	89.8	84.4	88.5	89.8	87.5	**90.2**
Average	31.4	41.1	43.6	46.2	43.1	46.8	46.3	48.8	45.2	44.3	49.2	52.7	**54.1**

Table 2

Accuracy (%) on Office-Caltech datasets using DeCaf6 features.

Task	Traditional Methods											Deep Methods					TFDF
	SVM	PCA	TCA	GFK	JDA	SCA	ARTL	JGSA	CORAL	DMM	CAPLS	AlexNet	DDC	DAN	DCORAL	DNN	
C → A	91.6	88.1	89.8	88.2	89.6	89.5	92.4	91.4	92.0	92.4	90.8	91.9	91.9	92.0	92.8	93.0	**93.5**
C → W	80.7	83.4	78.3	77.6	85.1	85.4	87.8	86.8	80.0	87.5	85.4	83.7	85.4	90.6	91.1	93.0	**95.6**
C → D	86.0	84.1	85.4	86.6	89.8	87.9	86.6	93.6	84.7	90.4	**95.5**	87.1	88.8	89.3	91.4	91.5	93.0
A → C	82.2	79.3	82.6	79.2	83.6	78.8	87.4	84.9	83.2	84.8	86.1	83.0	85.0	84.1	84.7	86.5	**87.8**
A → W	71.9	70.9	74.2	70.9	78.3	75.9	88.5	81.0	74.6	84.7	87.1	79.5	86.1	91.8	-	**94.9**	88.1
A → D	80.9	82.2	81.5	82.2	80.3	85.4	85.4	88.5	84.1	92.4	**94.9**	87.4	89.0	91.7	-	93.3	94.2
W → C	67.9	70.3	80.4	69.8	84.8	74.8	88.2	85.0	75.5	81.7	**88.2**	73.0	78.0	81.2	79.3	85.9	86.5
W → A	73.4	73.5	84.1	76.8	90.3	86.1	92.3	90.7	81.2	86.5	92.3	83.8	84.9	92.1	-	92.5	**93.2**
W → D	100.0	99.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98.7	100.0	100.0	100.0	100.0	-	100.0	98.7
D → C	72.8	71.7	82.3	71.4	85.5	78.1	87.3	86.2	76.8	83.3	**88.8**	79.0	81.1	80.3	82.8	83.1	87.7
D → A	78.7	79.2	89.1	76.3	91.7	90.0	92.7	92.0	85.5	90.7	93.0	87.1	89.5	90.0	-	93.3	**93.1**
D → W	98.3	98.0	99.7	99.3	99.7	98.6	**100.0**	99.7	99.3	99.3	**100.0**	97.7	98.2	98.5	-	99.2	98.0
Average	82.0	81.7	85.6	81.5	88.2	85.9	90.7	90.0	84.7	89.4	91.8	86.1	88.2	90.1	-	91.5	**92.5**

Table 3

Accuracy (%) on USPS+MNIST and COIL datasets.

Task	1NN	SVM	PCA	TCA	GFK	JDA	TJM	CORAL	SCA	ARTL	JGSA	McDA	MEDA	TFDF
U → M	44.7	62.2	45.0	51.2	46.5	59.7	52.3	30.5	48.0	67.7	68.2	-	72.1	**80.6**
M → U	65.9	68.2	66.2	56.3	61.2	67.3	63.3	49.2	65.1	88.8	80.4	-	89.5	**89.9**
COIL1 → COIL2	83.6	84.7	84.7	88.5	72.5	89.3	87.6	82.64	-	88.3	-	**94.1**	90.1	93.6
COIL2 → COIL1	82.8	82.9	84.0	85.8	74.2	88.5	87.4	82.36	-	84.0	-	89.6	87.1	**91.8**
Average	69.3	74.5	70.0	70.5	63.6	76.2	72.7	61.2	-	82.2	-	-	84.7	**89.0**

of datasets, it can convincingly verify that TFDF can build a robust adaptive classifier while reducing cross-domain discrepancy.

Secondly, both TFDF and McDA adopt MMCD to measure domain discrepancy, and they perform better than TCA, JDA, CORAL, and ARTL which either only consider the first-order statistical information or only consider the second-order statistical information. This improvement indicates that considering both the first-order and the second-order statistics simultaneously can capture more information for reducing cross-domain discrepancy. TFDF outperforms McDA because the proposed methods not only adopt MMCD to measure and decrease domain discrepancy but also consider the discriminative

information of features. Compared with MEDA, which also performs manifold regularization to learn local discriminative features, minimizing class confusion loss is helpful to learn global discriminative features, thus achieving better performance. Compared with CAPLS, which focus on pesudo label selection within the distribution alignment process, TFDF also achieves better performance although TFDF use all the pseudo labels (even with wrong pseudo labels), which shows the robustness of the proposed method. Moreover, the error matrixes of different algorithms are shown in Fig 1, which shows that TFDF can learn global discriminative information to avoid class confusion and get better performance.

Thirdly, TFDF also performs better than the deep methods (AlexNet, DDC, DAN, DCORAL, and DNN) on Office+Caltech10 datasets. Based on the powerful features extracted from deep models, some traditional methods such as DMM, MDSI-V, and TFDF can achieve better performance than deep methods. As we can see, TFDF achieves the best performance with a 1% improvement compared to DNN.

4.5. *Effectiveness Analysis*

4.5.1. *Ablation Study*

We conduct an ablation study to analyze how different components of our work contribute to the final performance. When learning the final classifier, TFDF involves four components: Structural Risk Minimization (SRM), Distribution Alignment (DA), Local Discriminative information (LD), and Global Discriminative information (GD). We empirically evaluate the importance of each component. To this end, we investigate different combinations of four components and report average classification accuracy on five datasets in Table 4. The first setting where only SRM is used is actually the source-only method where no adaptation is performed and this setting performs the worst. It can be observed that the accuracy is further improved after performing Distribution Alignment (DA). And the use of both global and local discriminative information improves the performance significantly on all datasets. Finally, a combination of all components can achieve the best results.

Table 4
Results of ablation study(accuracy (%))

Method				Office-Caltech (surf features)	COIL	MNIST-USPS
SRM	DA	LD	GD			
✓	×	×	×	49.93	82.22	54.83
✓	✓	×	×	51.92	90.29	74.89
✓	×	✓	×	51.60	83.26	56.50
✓	✓	✓	×	52.20	91.12	80.86
✓	✓	✓	✓	**54.10**	**92.71**	**85.22**

4.5.2. *Distribution Distance*

We run JDA, MEDA, and TFDF on task $C \rightarrow D$ (SURF features) using their optimal parameter settings. Then we compute the aggregate MMD distance and MMCD distance of each method on their induced embeddings by equation (5) and equation (9), respectively. To compute the true distance in both the marginal and conditional distributions across domains, we have to use the ground truth labels instead of the pseudo labels. However, the ground truth target labels are only used for verification, not for learning procedures.

Fig 3(a) and Fig 3(b) show the MMD distance and MMCD distance computed for each method and Fig 3(c) shows the prediction accuracy for each method. As we can see, MEDA and JDA can reduce the

(a) MMD distance w.r.t. #iterations (b) MMCD distance w.r.t. #iterations (c) Accuracy(%) w.r.t. #iterations
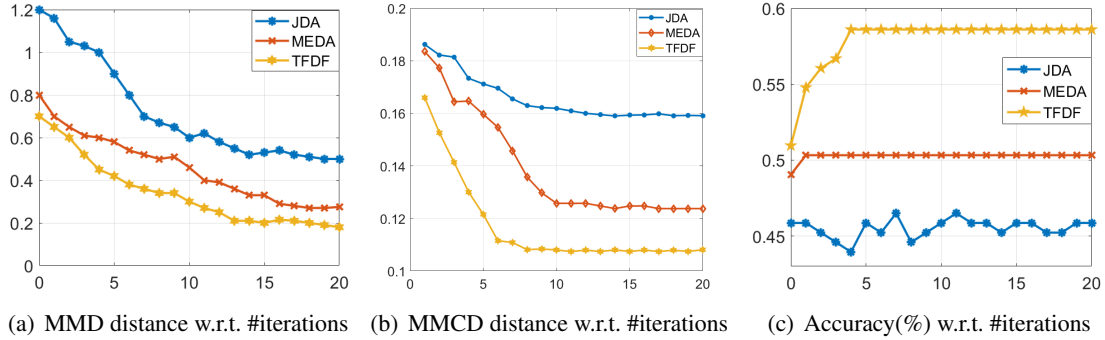
Fig. 3. MMD distance, classification accuracy on the task $C \to D$

domain discrepancy which can be measured by either MMD distance or MMCD distance and achieve good performance in the target domain while TFDF achieves better results. JDA only considers the first-order statistic, while TFDF adapts both the first-order and the second-order statistics. Besides, by minimizing class confusion, TFDF can learn more discriminative features and improve performance.

4.5.3. Feature Visualization.

In Fig 4, we visualize the feature representations of task USPS $\to$ MNIST (**U**$\to$**M**) (10 classes) by t-SNE[36] using JDA, MEDA and TFDF. Before adaptation, we can see that there is a large distribution discrepancy across domains. After adaptation, JDA learns invariant features which can reduce distribution discrepancy. MEDA further considers the dynamic factor between a marginal distribution and conditional distribution and makes a better adaptation. TFDF not only learns transferable features but also learns local and global discriminative features. Therefore, besides a small distribution discrepancy, the features in both domains are more discriminative and cab be easily classified by the classifier.

4.5.4. Time Complexity

We run JDA, MEDA, and TFDF on datasets Office-Caltech, COIL, and USPS+MNIST using their optimal parameter settings. Then we compute the running time of each method and the results are shown in Table 5. As we can see, JDA and MEDA need less time than TFDF as they are solved by a closed-form solution. Although TFDF-V can be solved by a closed-form solution, TFDF needs more iterations to get the final results than these two baselines, which is a shortcoming of the proposed method.

	Office-Caltech (surf features)	COIL	MNIST-USPS
JDA	5.66 min	0.94 min	8.08 min
MEDA	3.51 min	0.73 min	3.27 min
TFDF	33.10 min	7.86 min	52.42 min

Table 5

Runing time on different tasks

4.6. Parameter Sensitivity

In this section, we evaluate TFDF with a wide range of values for regularization parameters ρ, λ, η and neighbors number p. We only report the results on MNIST $\to$ USPS (M $\to$ U), COIL 1 $\to$ COIL2 and

(a) t-sne before adaptation   (b) t-sne by JDA   (c) t-sne by MEDA   (d) t-sne by TFDF
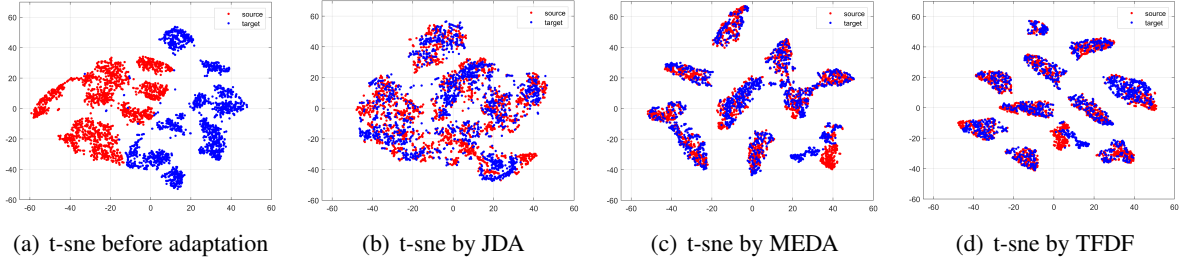
Fig. 4. Feature visualization on task $U \rightarrow M$

(a) neighbors p (b) ρ (c) λ (d) η
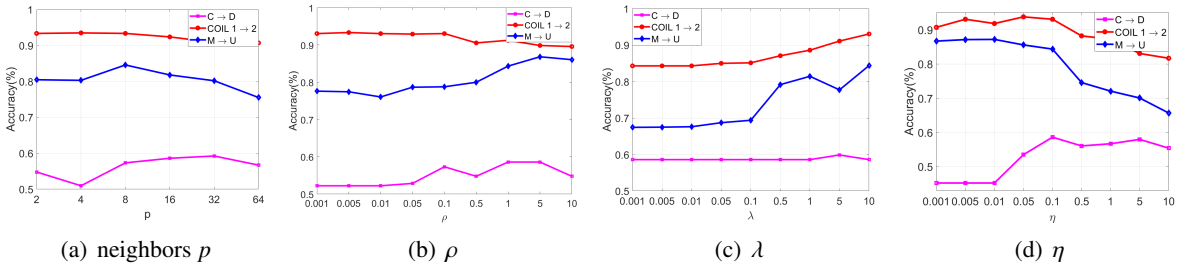
Fig. 5. Parameter sensitivity on three tasks.

$C \rightarrow D$ tasks, while similar trends on the other tasks are not shown due to space limitation. The results are shown in Fig 5. It can be observed that TFDF achieves a robust performance with regard to a wide range of parameter values. Specifically, $p \in [8, 16]$, $\rho \in [0.1, 1]$, $\lambda \in [1, 10]$ and $\eta \in [0.05, 0.1]$ are the optimal parameter values.

5. Conclusion

In this paper, we propose a method called *learning TransFerable and Discriminative Features for unsupervised domain adaptation* (TFDF), which could learn both transferable and discriminative features simultaneously. On the one hand, we adopt a recently proposed statistic called MMCD to measure domain discrepancy, which can capture both the first-order and the second-order statistical information simultaneously, thus more statistical information can be explored than MMD-based methods. On the other hand, we propose to learn both local and global discriminative features through manifold regularization and proposed class confusion loss respectively. With the principle of empirical risk minimization, TFDF also integrates the source classification error with the above objectives into a uniform optimization problem. Comprehensive experiments are conducted and the results verified the effectiveness of the proposed method.

6. Acknowledgement

References

[1] X. Wu, V. Kumar, J. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A.F.M. Ng, B. Liu, P.S. Yu, Z. Zhou, M. Steinbach, D. Hand and D. Steinberg, Top 10 algorithms in data mining, *Knowledge and Information Systems* **14** (2007), 1–37.

[2] S.J. Pan and Q. Yang, A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering* **22** (2010), 1345–1359.

[3] S.J. Pan, I.W.-H. Tsang, J.T. Kwok and Q. Yang, Domain Adaptation via Transfer Component Analysis, *IEEE Transactions on Neural Networks* **22** (2011), 199–210.

[4] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang and P.S. Yu, Visual Domain Adaptation with Manifold Embedded Distribution Alignment, in: *MM '18*, 2018.

[5] X. Chen, S. Wang, M. Long and J. Wang, Transferability vs. Discriminability: Batch Spectral Penalization for Adversarial Domain Adaptation, in: *ICML*, 2019.

[6] M. Long, J. Wang, G. Ding, S.J. Pan and P.S. Yu, Adaptation Regularization: A General Framework for Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering* **26** (2014), 1076–1089.

[7] M. Long, J. Wang, G. Ding, J.-G. Sun and P.S. Yu, Transfer Feature Learning with Joint Distribution Adaptation, *CVPR* (2013), 2200–2207.

[8] W. Dai, Q. Yang, G.-R. Xue and Y. Yu, Boosting for transfer learning, in: *ICML '07*, 2007.

[9] Y. Cao, M. Long and J. Wang, Unsupervised Domain Adaptation With Distribution Matching Machines, in: *AAAI*, 2018.

[10] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf and A.J. Smola, A Kernel Two-Sample Test, *J. Mach. Learn. Res.* **13** (2012), 723–773.

[11] J. Zhang, W. Li and P. Ogunbona, Joint Geometrical and Statistical Alignment for Visual Domain Adaptation, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5150–5158.

[12] L. Jing-jing, J. Mengmeng, L. Ke, Z. Lei and S.H. Tao, Locality Preserving Joint Transfer for Domain Adaptation, *arXiv: Computer Vision and Pattern Recognition* (2019).

[13] M. Belkin, P. Niyogi and V. Sindhwani, Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples, *J. Mach. Learn. Res.* **7** (2006), 2399–2434.

[14] B. Sun, J. Feng and K. Saenko, Return of Frustratingly Easy Domain Adaptation, in: *AAAI*, 2015.

[15] Y. Jin, X. Wang, M. Long and J. Wang, Less Confusion More Transferable: Minimum Class Confusion for Versatile Domain Adaptation, *ECCV* (2020).

[16] W. Zhang, X. Zhang, L. Lan and Z. Luo, Maximum Mean and Covariance Discrepancy for Unsupervised Domain Adaptation, *Neural Processing Letters* **51** (2019), 347–366.

[17] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, *ICLR* (2014).

[18] A.J. Wyner, M. Olson, J. Bleich and D. Mease, Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers, *J. Mach. Learn. Res.* **18** (2017), 48:1–48:33.

[19] M. Jing, J. Li, J. Zhao and K. Lu, Learning Distribution-Matched Landmarks for Unsupervised Domain Adaptation, in: *DASFAA*, 2018.

[20] J. Wang, Y. Chen, S. Hao, W. Feng and Z. Shen, Balanced Distribution Adaptation for Transfer Learning, *2017 IEEE International Conference on Data Mining (ICDM)* (2017), 1129–1134.

[21] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko and T. Darrell, Deep Domain Confusion: Maximizing for Domain Invariance, *ArXiv* **abs/1412.3474** (2014).

[22] A. Krizhevsky, I. Sutskever and G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: *NIPS*, 2012.

[23] M. Long, Y. Cao, J. Wang and M.I. Jordan, Learning Transferable Features with Deep Adaptation Networks, *ICML* (2015).

[24] B. Sun and K. Saenko, Deep CORAL: Correlation Alignment for Deep Domain Adaptation, in: *ECCV Workshops*, 2016.

[25] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger and S. Saminger-Platz, Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning, in: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. https://openreview.net/forum?id=SkB-_mcel.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville and Y. Bengio, Generative Adversarial Networks, *ICLR* (2014).

[27] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand and V. Lempitsky, Domain-Adversarial Training of Neural Networks, *J. Mach. Learn. Res.* **17** (2016), 59:1–59:35.

[28] E. Tzeng, J. Hoffman, K. Saenko and T. Darrell, Adversarial Discriminative Domain Adaptation, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 2962–2971.

[29] K. Saito, K. Watanabe, Y. Ushiku and T. Harada, Maximum Classifier Discrepancy for Unsupervised Domain Adaptation, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 3723–3732.

[30] K. Fukunaga, Introduction to Statistical Pattern Recognition, 1972.

[31] S. Li, S. Song, G. Huang, Z. Ding and C. Wu, Domain Invariant and Class Discriminative Feature Learning for Visual Domain Adaptation, *IEEE Transactions on Image Processing* **27** (2018), 4260–4273.

[32] C. Chen, Z. Chen, B. Jiang and X. Jin, Joint Domain Alignment and Discriminative Feature Learning for Unsupervised Deep Domain Adaptation, in: *AAAI*, 2019.

[33] O. Bousquet, Statistical Learning Theory, 2003.

[34] B. Schölkopf, R. Herbrich and A.J. Smola, A Generalized Representer Theorem, in: *COLT/EuroCOLT*, 2001.

[35] B. Gong, Y. Shi, F. Sha and K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), 2066–2073.

[36] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, in: *ICML*, 2014.

[37] M. Long, J. Wang, G. Ding, J.-G. Sun and P.S. Yu, Transfer Joint Matching for Unsupervised Domain Adaptation, *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1410–1417.

[38] M. Ghifary, D. Balduzzi, W.B. Kleijn and M. Zhang, Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** (2015), 1414–1430.

[39] Q. Wang, P. Bu and T. Breckon, Unifying Unsupervised Domain Adaptation and Zero-Shot Visual Recognition, *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), 1–8.

[40] C. Wu, W. Wen, T. Afzal, Y. Zhang, Y. Chen and H.H. Li, A Compact DNN: Approaching GoogLeNet-Level Accuracy of Classification and Domain Adaptation, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 761–770.

[41] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F.C. Pereira and J.W. Vaughan, A theory of learning from different domains, *Machine Learning* **79** (2009), 151–175.

[42] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, *(CVPR)* (2016), 770–778.

[43] N. Smith and M.J.F. Gales, Speech Recognition using SVMs, in: *NIPS*, 2001.

[44] X. Glorot, A. Bordes and Y. Bengio, Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach, in: *ICML*, 2011.

[45] C.S. Perone, P.L. Ballester, R.C. Barros and J. Cohen-Adad, Unsupervised domain adaptation for medical imaging segmentation with self-ensembling, *NeuroImage* **194** (2018), 1–11.

[46] G.S. Griffin, A. Holub and P. Perona, Caltech-256 Object Category Dataset, 2007.

[47] Q. Wang and T.P. Breckon, Unsupervised Domain Adaptation via Structured Prediction Based Selective Pseudo-Labeling, *AAAI* (2020).

Appendix A. Proof of theorem 1

According to [16], we can approximate the convex upper bound of the second $||\beta^T K Z_0 K^T \beta||_F^2$ and the fourth $||\beta^T K Z_c K^T \beta||_F^2$ terms by using the following theorem:

Theorem 2. *given the constraint that $\beta^T K H K^T \beta = I$, the following inequality holds*

$$||\beta^T K Z_c K^T \beta||_F^2 \leqslant \sigma k ||\beta^T K Z_c K^T||_F^2 \tag{26}$$

where k is the feature dimensionality and $\sigma = ||(KHK)^{-\frac{1}{2}}||^2$.

Proof.

$$||\beta^T K Z_c K^T \beta||_F^2 = ||\beta^T K Z_c K^T (KHK^T)^{-1/2} (KHK^T)^{1/2} \beta||_F^2$$
$$\leqslant ||\beta^T K Z_c K^T (KHK^T)^{-1/2}||_F^2 ||(KHK^T)^{1/2} \beta||_F^2$$
$$= k ||\beta^T K Z_c K^T (KHK^T)^{-1/2}||_F^2$$
$$\leqslant k ||\beta^T K Z_c K^T||_F^2 ||(KHK^T)^{-1/2}||_F^2$$
$$= k\sigma ||\beta^T K Z_c K^T||_F^2$$

□

The first equation holds because KHK^T is semi-definite positive, while the second and the fourth inequalities follow the Cauchy-Schwarz inequality. In terms of the constraint of theorem 1, $k = \text{tr}(\beta^T KHK^T \beta) = \text{tr}(I_k)$.