Discriminative Transfer Feature and Label Consistency for Cross-Domain Image Classification

Shuang Li, Chi Harold Liu, *Senior Member, IEEE*, Limin Su, Binhui Xie, Zhengming Ding, *Member, IEEE*, C. L. Philip Chen, *Fellow, IEEE*, and Dapeng Wu, *Fellow, IEEE*

***Abstract*— Visual domain adaptation aims to seek an effective transferable model for unlabeled target images by benefiting from the well-labeled source images following different distributions. Many recent efforts focus on extracting domain-invariant image representations via exploring target pseudo labels, predicted by the source classifier, to further mitigate the conditional distribution shift across domains. However, two essential factors are overlooked by most existing methods: 1) the learned transferable features should be not only domain invariant but also category discriminative; and 2) the target pseudo label is a two-edged sword to cross-domain alignment. In other words, the wrongly predicted target labels may hinder the class-wise domain matching. In this article, to address these two issues simultaneously, we propose a discriminative transfer feature and label consistency (DTLC) approach for visual domain adaptation problems, which can naturally unify cross-domain alignment with discriminative information preserved and label consistency of source and target data into one framework. To be specific, DTLC first incorporates class discriminative information by penalizing the maximum distance of data pair in the same class and the minimum distance of data pair sharing the different labels for each data into the distribution alignment of both domains. The target pseudo labels are then refined based on the label consistency within the domains. Thus, the transfer feature learning and coarse-to-fine target labels would be coupled to benefit each other in an iterative way. Comprehensive experiments on several visual cross-domain benchmarks verify that DTLC can gain remarkable margins over state-of-the-art (SOTA) nondeep visual domain adaptation methods and even be comparable to competitive deep domain adaptation ones.**

***Index Terms*— Cross-domain image classification, discriminative transfer feature learning, label consistency, visual domain adaptation.**

Manuscript received May 23, 2019; revised October 31, 2019 and November 22, 2019; accepted December 3, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61902028 and in part by the National Key Research and Development Plan of China under Grant 2018YFB1003701 and Grant 2018YFB1003700. *(Corresponding author: Chi Harold Liu.)*

S. Li, C. H. Liu, L. Su, and B. Xie are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: shuangli@bit.edu.cn; liuchi02@gmail.com; sulimin@bit.edu.cn; binhuixie@bit.edu.cn).

Z. Ding is with the Department of Computer, Information and Technology, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: zd2@iu.edu).

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: philip.chen@ieee.org).

D. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6130 USA (e-mail: dpwu@ieee.org).

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

Digital Object Identifier 10.1109/TNNLS.2019.2958152

I. Introduction

PAST decades have witnessed a great success of supervised learning paradigms on numerous real-world applications [1]–[5]. However, the reliance on sufficiently annotated training data and the common assumption that training and test data are drawn from the same distribution may constitute limitations for achieving ideal generalization ability of standard supervised learning methods on massive cross-domain vision tasks [6]–[11]. Hence, it is of vital importance to leverage a well-labeled source domain to facilitate designing effective models when our interested target domain lies in a different but related distribution. To this end, domain adaptation is explored to mitigate the distribution gap by utilizing the latent domain-invariant knowledge across domains, and it has been proven to be a promising technique to tackle these domain mismatching problems with considerable research efforts [9], [12]–[19].

Most recent domain adaptation works focus on either reweighting source data to derive a weighted classifier for target data [13], [20]–[22] or extracting domain-invariant features to mitigate the discrepancy between source and target such that useful source knowledge can be transferred to target domain effectively [7], [10], [23], [24]. Though the feature extraction strategy is widely explored and has achieved appealing performance, the learned features are often distorted by minimizing a distance metric to characterize the discrepancy of source and target distributions, such as maximum mean discrepancy (MMD) [25]. Feature distortion may damage the intrinsic class-wise structures transferred from source to target, which degrades the target recognition performance [26]. Thus, we should enforce the learned features to be not only domain invariant, which ensures the distributions of both domains to be well aligned, but also category discriminative, which will facilitate achieving high target image classification accuracy and compensate the side effect of feature distortion.

To this end, Baktashmotlagh *et al.* [23] encouraged the embedded features to minimize the intradomain variance by introducing the class clustering in source data. Furthermore, [26] aimed to improve the discriminativeness of learned representations by simultaneously maximizing the interclass dispersion and minimizing the intraclass scatter. Different from them, inspired by the triplet loss in [27], we propose a straightforward but effective loss term to penalize the maximum distance of data pair in the same class and the minimum distance of data pair sharing the different labels for each of the data in the source and target domains, respectively. This

mechanism targets at discovering and optimizing the hardest data pairs in both domains and manages to minimize the distance of positive pairs while maximizing the distance of negative pairs in the learned feature space.

In addition, many recent feature extraction-based methods incorporate target pseudo labels, mostly only predicted by the source classifier, to further alleviate the class-wise distribution differences of two domains. This strategy indeed boosts the final target recognition performance by exploring the class-wise adaptation [10], [28]. However, it is a two-edged sword to explore target pseudo labels for cross-domain alignment, and this issue has always been overlooked. On the one hand, when the pseudo labels are highly accurate, the marginal and conditional distributions across the domains will be perfectly matched. Then, the transfer feature learning procedure and target label prediction will facilitate each other in a positive cycle. On the other hand, target pseudo labels are usually predicted by the source classifier trained on the projected source data. This scheme is likely to be overfitting to the source data, resulting in many incorrect target predictions. These wrongly predicted target labels will mislead the class-wise distribution alignment and result in performance degradation accordingly. Thus, guaranteeing the correctness of target pseudo labels is crucial to learn ideal domain-invariant features. Considering that most existing methods ignore to optimize the feature learning and refine the target pseudo labels in one framework, they are unlikely to benefit from each other to enhance the final performance from an iterative learning perspective.

To alleviate the side effect of inaccurate target pseudo labels, Hou *et al.* [28] managed to refine target pseudo labels by classic label propagation techniques [29] after the whole feature learning procedure. Differently, we exploit the label consistency of source and target data from between- and within-domain perspectives to refine target pseudo labels iteratively. To be specific, we first explore the label consistency across the domains. Intuitively, the labels of target data that are closer to the source domain are more consistent with source labels. We resort to the proposed class-wise domain classifier to measure the distance between each target sample and source domain. Then, all the target predictions obtained by the source classifier can be reweighted according to the distances. Finally, incorporating the structural consistency within the target domain, target pseudo labels will be updated from target data with large weights to small weights gradually. Furthermore, they will feedback to the feature learning step in the next iteration.

As noted earlier, in this article, we propose a discriminative transfer feature learning and label consistency (DTLC) approach to address visual domain adaptation problems. As shown in Fig. 1, DTLC incorporates domain-invariant feature learning with class-discriminative information preserved and coarse-to-fine target label refinement leveraging label consistency into one general framework. These two procedures can assist each other adaptively for effective knowledge transfer. To sum up, we list our contributions in fourfolds.

1) We propose a novel approach to address two critical issues in cross-domain image classification simultaneously. First, the learned domain-invariant representations should be equipped with category-discriminative knowledge to compensate for the side effect of feature distortion. Second, target pseudo labels should be refined to explore the valuable class structures under the target data. The extensive experimental results in Section IV explicitly verify that these two procedures are complementary to each other.
2) For discriminative transfer feature learning, we propose an intuitive but effective loss to optimize the distances of hardest data pairs, which minimizes intraclass compactness and maximizes interclass dispersion across two domains. The optimal projection in each iteration can be solved closely via a generalized eigenvalue problem.
3) For target pseudo label refinement, we explore the label consistency of source and target data from between-/within-domain perspectives, by leveraging the proposed class-wise domain classifier and geometric information.
4) Extensive experimental results on several popular visual cross-domain benchmarks, including CMU-PIE, Office-31 (DeCAF$_7$ and ResNet-50), Office-Home (ResNet-50), Office+Caltech10 (SURF and DeCAF$_6$), and ImagetNet+VOC2007, have demonstrated the superiority of DTLC to other state-of-the-art (SOTA) domain adaptation methods. Particularly, for CMU-PIE, the average classification accuracy of DTLC outperforms the best baseline [26] 12.7%. Also, when utilizing deep features, DTLC is also comparable to SOTA deep methods.

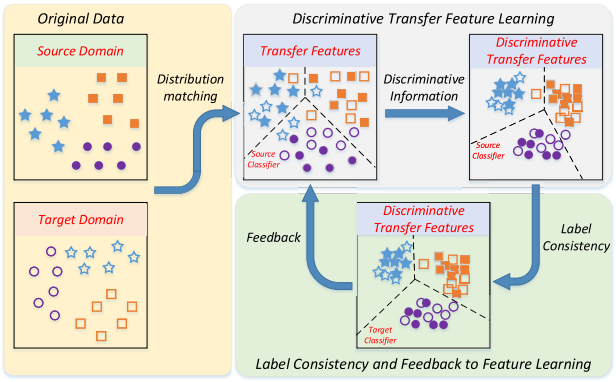


Fig. 1. Overview of the proposed DTLC. In the discriminative transfer feature learning stage, DTLC simultaneously minimizes the domain discrepancy between both domains and enhances the class-wise discriminability of the learned representations. In the label consistency stage, DTLC will refine target pseudo labels based on the proposed class-wise domain classifier and target structural information. Therefore, the misclassified target pseudo labels can be refined according to the label consistency between and within domains, and the refined target labels will feedback to the next iteration of transfer feature learning. Finally, the discriminative transfer feature learning and target label refinement can be coupled to boost each other in an iterative way.

The rest of this article is organized as follows. Section II will first review the related domain adaptation methods. In Section III, the proposed DTLC is introduced in detail. We conduct empirical studies on extensive benchmarks and analytical experiments in Section IV. Section V concludes this article.

II. Related Works

Domain adaptation has achieved a series of promising advances in recent years [10], [18], [19], [30]. In this section, we mainly review the related domain adaptation methods from two aspects: shallow adaptation [1], [7], [21], [26] and deep

adaptation [3], [18], [19], [30]. Moreover, their differences with the proposed DTLC will be highlighted particularly.

A. Shallow Domain Adaptation Methods

Generally, shallow domain adaptation methods can be roughly divided into two main categories: instance reweighting-based methods and feature extraction-based methods [1]. Instance reweighting methods focus on assigning different weights to source data such that the discrepancy between two domains can be minimized [20], [21]. For instance, [21] proposed a kernel mean matching (KMM) approach to decrease the distance of source and target means in a reproducing kernel Hilbert space (RKHS) by reweighting each source data. Li *et al.* [22] leveraged the domain classifier to measure the closeness of each data to the other domain. Then, the weight of each of the data can be calculated accordingly. However, these methods usually assume that source and target data follow the identical conditional distribution, which cannot be held when the domain difference is substantially large.

Another main category of domain adaptation is the feature extraction method, which aims to derive domain-invariant features or latent subspaces by reducing the distribution difference across domains [8]–[10], [31], [32]. To be specific, Gong *et al.* [8] proposed a geodesic flow kernel (GFK) method, which embeds both domains into the Grassmann manifold and models the domain shift by constructing geodesic flows. CORAL [31] explores the second-order statistics of source and target to align coupled domain subspaces. Scatter component analysis (SCA) [32] takes the between- and within-class scatters into consideration to further minimize the divergence of source and target. Different from them, transfer component analysis (TCA) is proposed to learn transfer components in RKHS to minimize the distance of the source and target marginal distributions using the MMD metric [9]. Later on, leveraging target pseudo labels, JDA [10], proposes to match both marginal and conditional distributions of two domains by requiring their total means and class means to be close to each other. Furthermore, label and structure consistency (LSC) proposed in [28] leverages label propagation [29] for adaption purpose. Other works exploit the techniques of sparse representation [33] and geometrical alignment [7] to achieve better performance.

To explore the discriminative information of both domains, [23] incorporated the supervised source class clustering into the interdomain MMD minimization objective. Besides, domain-irrelevant class clustering (DICE) [34] proposes a clustering-promoting technology to develop an ensemble model and gives the final decision on unlabeled target data via majority voting. Furthermore, domain-invariant and class discriminative (DICD) [26] proposes to simultaneously maximize the interclass dispersion and minimize the intraclass variance of source and target data. Note that, explicitly different from them, our DTLC aims to explore the hardest data pairs across the domains in a lightweight way rather than calculating all the distances between source and target samples. These related methods only focus on deriving transferable features with discrimination across domains, while the label consistency is also crucial. The proper exploitation of label consistency between and within domains could benefit the transfer feature learning a lot. Moreover, the appropriate integration of transfer feature learning and target label refinement into one training procedure will dramatically improve the performance.

Apart from the aforementioned feature extraction-based methods, transfer joint machine (TJM) in [24] jointly aligns both domain features and reweights source data to reduce the domain discrepancy. Ding and Fu [14] proposed a robust transfer metric learning (RTML) approach to mitigate the domain shift in two directions consisting of sample space and feature space. Courty *et al.* [35] addressed the domain shift problems based on optimal transport strategies. Lu *et al.* [36] presented an linear discriminant analysis (LDA)-inspired DA (LDADA), which only leverages the class mean to learn class-wise linear projections. Although these methods have shown their effectiveness to solve the cross-domain problems, they ignore to simultaneously take the transfer feature learning and label inference into consideration. By contrast, DTLC could effectively incorporate them into one framework and optimize them iteratively to benefit each other in a positive feedback way.

B. Deep Domain Adaptation Methods

More recently, deep domain adaptation methods embed the domain alignment into the pipeline of deep learning to jointly mitigate the domain discrepancy and derive a transferable classifier [17]–[19], [30], [34], [37], [38]. Deep domain confusion (DDC) [39] first introduces an MMD-based domain confusion loss into CNN architecture. Following this idea, domain adaptation network (DAN) [18] embeds all hidden representations of task-specific layers in an RKHS where the source and target means are explicitly matched. Besides mitigating the domain shift across domains, residual transfer network (RTN) [17] also conducts the classifier adaptation across domains. To further learn discriminative prototype representations, Pinheiro [38] learned a pair of similarity functions to perform the classification by calculating the similarity between the feature representations of each class.

Different from them, Ganin and Lempitsky [30] novelly embedded adversarial learning into deep networks by introducing a gradient reversal layer to reduce distribution discrepancy. Based on this framework, Tzeng *et al.* [40] developed an approach by exploring an adversarial loss in aligning both domains in terms of feature level. Pei *et al.* [41] trained multiple class-wise domain discriminators according to the number of classes. Recently, Satio *et al.* [19] introduced a new adversarial scheme by learning the two classifiers' discrepancies. However, a relatively long training time, as well as the resources consumption, is still big challenges for deep methods.

Moreover, many recent shallow domain adaptation works [7], [28] have demonstrated to achieve comparable performance to deep methods based on general deep features. This indicates generic deep feature learning, and domain alignment could be in separate ways. Thus, we equip domain alignment over generic deep features by further coupling domain mismatch and preserving the discriminative information across domains. Simultaneously, label consistency between and within domains is also taken into consideration such that the target pseudo labels can be refined in each iteration. As a feedback, the refined target labels will be

unified into the domain-invariant feature learning to facilitate the alignment.

III. Proposed Framework

A. Problem Definition and Motivation

In unsupervised domain adaptation, labeled source domain $\mathcal{D}_s = \{(\boldsymbol{x}_{si}, y_{si})\}_{i=1}^{n_s} = \{\mathbf{X}_s, \boldsymbol{y}_s\}$ and unlabeled target domain $\mathcal{D}_t = \{\boldsymbol{x}_{tj}\}_{j=1}^{n_t} = \{\mathbf{X}_t\}$ are given, where n_s and n_t are the number of source and target samples, respectively. Assume $\boldsymbol{x}_{si}, \boldsymbol{x}_{tj} \in \mathcal{X}$, and $\mathcal{X} \subset \mathbb{R}^m$ is the feature space. Correspondingly, $y_{si} \in \mathcal{Y}$ is the label of $\boldsymbol{x}_{si}$, and $\mathcal{Y} \subset \mathbb{R}$ is the label space. Due to the domain shift, the marginal distributions $P_s(\boldsymbol{x}_s) \neq P_t(\boldsymbol{x}_t)$, and the conditional distributions $Q_s(y_s|\boldsymbol{x}_s) \neq Q_t(y_t|\boldsymbol{x}_t)$, we first devote to seeking an effective transformation $\phi(\cdot)$ such that the domain discrepancy between the domains can be minimized, i.e., $P_s(\phi(\boldsymbol{x}_s)) \approx P_t(\phi(\boldsymbol{x}_t))$ and $Q_s(y_s|\phi(\boldsymbol{x}_s)) \approx Q_t(y_t|\phi(\boldsymbol{x}_t))$ with discriminative structures preserved as much as possible. Meanwhile, to further align the class-wise distributions across domains, the target pseudo labels should be refined simultaneously by exploring the intrinsic label consistency between and within domains.

Recently, to effectively mitigate the domain discrepancy, most existing methods focus on learning domain-invariant features across domains by leveraging target pseudo labels, which are just predicted by the learned source classifier [7], [10], [26]. However, the source classifier trained in this manner may be overfitting to the embedded source data. Since the existing inevitable domain shift, those target data, who are dissimilar to the source domain, would be incorrectly inferred. Consequently, the wrong predicted labels may degrade the class-wise distributions alignment and cause the negative transfer. Thus, to explicitly refine target pseudo labels is crucial for effective adaptation, and the refined target pseudo labels will definitely facilitate distribution matching in a positive feedback way. Moreover, as we all know that, the discriminative information is critical to standard classification problems [42], [43]. Hence, we expect the embedded features to be not only domain invariant but also category discriminative.

To solve this dilemma, we incorporate discriminative transfer feature learning and target label refinement by label consistency into one general framework. The transfer feature learning step will explicitly minimize the MMD distances of both marginal and conditional distributions across domains and preserve the data category-discriminative knowledge by minimizing the intraclass dispersion and maximizing the interclass compactness. Simultaneously, in the learned embedding subspace, target pseudo labels will be refined according to the label consistency between and within the source and target domains to benefit the transfer feature learning in an effective way. Following this line, these two procedures are subtly incorporated into DTLC and boost each other adaptively.

B. Discriminative Transfer Feature Learning

1) Domain-/Class-Wise Alignment Revisit: Above all, we expect the learned representations to be domain invariant such that useful knowledge can be transferred from source to target effectively. MMD criteria [25] are widely used to measure the distance between two distributions [10], [26], [44]. Here, to reduce the distance between marginal and conditional distributions of source and target, we attempt to seek a linear domain invariant projection $\mathbf{P}$ to the couple $\mathbf{X}_s$ and $\mathbf{X}_t$ for both domain- and class-wise adaptations. The projected representations of source and target data are formulated as $\boldsymbol{z}_s = \mathbf{P}^\top \boldsymbol{x}_s$ and $\boldsymbol{z}_t = \mathbf{P}^\top \boldsymbol{x}_t$.

First of all, domain-wise adaptation is to minimize the distance between sample means of two domains in the d-dimensional embedding subspace by using the empirical MMD. Second, class-wise adaptation is to minimize the distance of conditional distributions between source and target, and the class-wise adaption loss is defined with the target pseudo labels predicted by the source classifier. Following previous work in [10], we could formulate the whole MMD loss as

$$\begin{aligned}\mathcal{J}_{mmd} &= \sum_{c=0}^{C} \left\| \frac{1}{n_s^{(c)}} \sum_{\boldsymbol{x}_{si} \in \mathcal{D}_s^{(c)}} \boldsymbol{z}_{si}^{(c)} - \frac{1}{n_t^{(c)}} \sum_{\boldsymbol{x}_{tj} \in \hat{\mathcal{D}}_t^{(c)}} \boldsymbol{z}_{tj}^{(c)} \right\|^2 \\ &= \sum_{c=0}^{C} \mathrm{Tr}(\mathbf{P}^\top \mathbf{X} \mathbf{W}_c \mathbf{X}^\top \mathbf{P}) = \mathrm{Tr}(\mathbf{P}^\top \mathbf{X} \mathbf{W} \mathbf{X}^\top \mathbf{P}) \end{aligned} \tag{1}$$

where $\mathbf{W} = \sum_{c=0}^{C} \mathbf{W}_c$ and $c = 0$ means the data from the whole domain, i.e., $n_s^{(0)} = n_s$, $n_t^{(0)} = n_t$, $\mathcal{D}_s^{(0)} = \mathcal{D}_s$, and $\mathcal{D}_t^{(0)} = \mathcal{D}_t$. $\mathcal{D}_s^{(c)}$ and $\hat{\mathcal{D}}_t^{(c)}$ denote all the source and target samples with their true and pseudo class labels being c. $n_s^{(c)}$ and $n_t^{(c)}$ are the numbers of source and target samples in (pseudo) class c, respectively. $\mathbf{W}_c$ is the class-conditional MMD matrix for class c, which can be computed as

$$\mathbf{W}_c = \begin{bmatrix} \frac{1}{n_s^{(c)} n_s^{(c)}} \mathbf{1}_{n_s}^{(c)} \mathbf{1}_{n_s}^{(c)\top} & -\frac{1}{n_s^{(c)} n_t^{(c)}} \mathbf{1}_{n_s}^{(c)} \mathbf{1}_{n_t}^{(c)\top} \\ -\frac{1}{n_s^{(c)} n_t^{(c)}} \mathbf{1}_{n_t}^{(c)} \mathbf{1}_{n_s}^{(c)\top} & \frac{1}{n_t^{(c)} n_t^{(c)}} \mathbf{1}_{n_t}^{(c)} \mathbf{1}_{n_t}^{(c)\top} \end{bmatrix}. \tag{2}$$

Here, the dimensions of $\mathbf{1}_{n_s}^{(c)}$ and $\mathbf{1}_{n_t}^{(c)}$ are n_s and n_t, respectively. Each element of $\mathbf{1}_{n_s}^{(c)}$ and $\mathbf{1}_{n_t}^{(c)}$ is defined as

$$\left(\mathbf{1}_{n_s}^{(c)}\right)_i = \begin{cases} 1, & \boldsymbol{x}_i \in \mathcal{D}_s^{(c)} \\ 0, & \text{otherwise,} \end{cases} \quad \left(\mathbf{1}_{n_t}^{(c)}\right)_j = \begin{cases} 1, & \boldsymbol{x}_j \in \hat{\mathcal{D}}_t^{(c)} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Clearly, if we minimize the empirical MMD loss $\mathcal{J}_{mmd}$, the disparity between source and target could be mitigated effectively such that the learned low-dimensional representations for source and target will be domain invariant.

2) Category-Discriminative Information Preservation: Minimizing (1) can align source and target distributions in the learned feature space but cannot guarantee the learned representations to be discriminative enough for the classification task. To avoid the feature distortion caused by the distribution alignment, and benefit the discriminative knowledge transfer from source to target, we expect the transformed data to be close when they belong to the same class and to be far away when they are with nonmatching labels.

Inspired by the triplet loss [27], [45], which minimizes the distance of the within-class sample pairs while maximizing the distance of the between-class sample pairs in feature embedding space, we only focus on the hardest sample pairs to optimize as in Fig. 2, which is more efficient and effective. In other words, DTLC proposes to select the most dissimilar sample pair with the same label to minimize their difference and choose the most similar sample pair with different labels

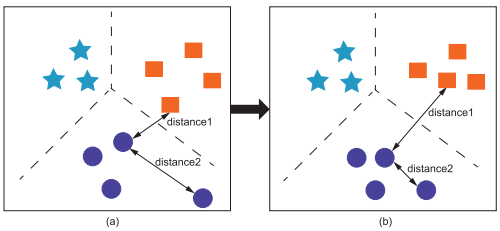


Fig. 2. Illustration of the hardest case distance optimization. For each sample, we will minimize the distance between itself and the farthest sample in the same class and simultaneously maximize the distance between itself and the nearest sample with different labels. (a) Original and (b) optimized distances between samples.

to maximize their distance. Thus, for each data in source and target, DTLC only concentrates on the hardest cases.

To be specific, for each sample in the source domain, we can find out the corresponding farthest sample with same label and the nearest sample with different labels. Then, the distance loss term and its matrix formulation can be represented as

$$\begin{aligned}
&\mathcal{J}_{s,distance} \\
&= \sum_{c=1}^{C} \sum_{\boldsymbol{x}_{si} \in \mathcal{D}_s^{(c)}} \underbrace{\max_{\boldsymbol{x}_{sj} \in \mathcal{D}_s^{(c)}} \|z_{si} - z_{sj}\|^2}_{\text{to shorten}} - \underbrace{\min_{\boldsymbol{x}_{sk} \notin \mathcal{D}_s^{(c)}} \|z_{si} - z_{sk}\|^2}_{\text{to enlarge}} \\
&= \sum_{c=1}^{C} \mathrm{Tr}\big(\mathbf{P}^\top \mathbf{X}_s \big(\mathbf{M}_{s,same}^{(c)} - \mathbf{M}_{s,diff}^{(c)}\big) \mathbf{X}_s^\top \mathbf{P}\big). \qquad (4)
\end{aligned}$$

Each element at the ith row and jth column of matrix $\mathbf{M}_{s,same}^{(c)}$ and $\mathbf{M}_{s,diff}^{(c)}$ can be calculated as

$$\begin{aligned}
&\big(\mathbf{M}_{s,same}^{(c)}\big)_{ij} \\
&= \begin{cases}
I\big(\boldsymbol{x}_{si} \in \mathcal{D}_s^{(c)}\big) \\
\quad + \displaystyle\sum_{\boldsymbol{x}_{sq} \in \mathcal{D}_s^{(c)}} I\left(\boldsymbol{x}_{si} = \arg\max_{\boldsymbol{x}_{sk} \in \mathcal{D}_s^{(c)}} \|z_{sq} - z_{sk}\|^2\right), & i = j \\
-I\left(\boldsymbol{x}_{sj} \in \mathcal{D}_s^{(c)}, \boldsymbol{x}_{si} = \arg\max_{\boldsymbol{x}_{sk} \in \mathcal{D}_s^{(c)}} \|z_{sj} - z_{sk}\|^2\right) \\
-I\left(\boldsymbol{x}_{si} \in \mathcal{D}_s^{(c)}, \boldsymbol{x}_{sj} = \arg\max_{\boldsymbol{x}_{sk} \in \mathcal{D}_s^{(c)}} \|z_{si} - z_{sk}\|^2\right), & i \neq j
\end{cases} \qquad (5)
\end{aligned}$$

and

$$\begin{aligned}
&\big(\mathbf{M}_{s,diff}^{(c)}\big)_{ij} \\
&= \begin{cases}
I\big(\boldsymbol{x}_{si} \in \mathcal{D}_s^{(c)}\big) \\
\quad + \displaystyle\sum_{\boldsymbol{x}_{sq} \in \mathcal{D}_s^{(c)}} I\left(\boldsymbol{x}_{si} = \arg\min_{\boldsymbol{x}_{sk} \notin \mathcal{D}_s^{(c)}} \|z_{sq} - z_{sk}\|^2\right), & i = j \\
-I\left(\boldsymbol{x}_{sj} \in \mathcal{D}_s^{(c)}, \boldsymbol{x}_{si} = \arg\min_{\boldsymbol{x}_{sk} \notin \mathcal{D}_s^{(c)}} \|z_{sj} - z_{sk}\|^2\right) \\
-I\left(\boldsymbol{x}_{si} \in \mathcal{D}_s^{(c)}, \boldsymbol{x}_{sj} = \arg\min_{\boldsymbol{x}_{sk} \notin \mathcal{D}_s^{(c)}} \|z_{si} - z_{sk}\|^2\right), & i \neq j
\end{cases} \qquad (6)
\end{aligned}$$

where $I(\cdot)$ is an indicator function. If we denote $\mathbf{M}_{s,same} = \sum_{c=1}^{C} \mathbf{M}_{s,same}^{(c)}$ and $\mathbf{M}_{s,diff} = \sum_{c=1}^{C} \mathbf{M}_{s,diff}^{(c)}$, then the distance loss for all the source samples is rewritten as

$$\mathcal{J}_{s,distance} = \mathrm{Tr}\big(\mathbf{P}^\top \mathbf{X}_s (\mathbf{M}_{s,same} - \mathbf{M}_{s,diff}) \mathbf{X}_s^\top \mathbf{P}\big). \qquad (7)$$

Similar derivation and definitions can also be applied to the target data according to their pseudo labels. Thus, the distance loss term for the target domain can be formulated as

$$\mathcal{J}_{t,distance} = \mathrm{Tr}\big(\mathbf{P}^\top \mathbf{X}_t (\mathbf{M}_{t,same} - \mathbf{M}_{t,diff}) \mathbf{X}_t^\top \mathbf{P}\big). \qquad (8)$$

Define $\mathbf{M}_{same} = \mathrm{diag}(\mathbf{M}_{s,same}, \mathbf{M}_{t,same})$ and $\mathbf{M}_{diff} = \mathrm{diag}(\mathbf{M}_{s,diff}, \mathbf{M}_{t,diff})$, and we can obtain the whole distance loss term for both domains as

$$\begin{aligned}
\mathcal{J}_{distance} &= \mathcal{J}_{s,distance} + \mathcal{J}_{t,distance} \\
&= \mathrm{Tr}(\mathbf{P}^\top \mathbf{X} (\mathbf{M}_{same} - \mathbf{M}_{diff}) \mathbf{X}^\top \mathbf{P}) \\
&= \mathrm{Tr}(\mathbf{P}^\top \mathbf{X} \mathbf{M} \mathbf{X}^\top \mathbf{P}). \qquad (9)
\end{aligned}$$

The minimization of (9) encourages small intraclass compactness and large interclass dispersion, thereby enhancing the discriminative power of the learned representations.

3) Overall Formulation: We formulate the loss function of the transfer feature learning procedure in DTLC by incorporating (1) and (9) as

$$\mathcal{J} = \mathcal{J}_{mmd} + \alpha \mathcal{J}_{distance} + \beta \|\mathbf{P}\|_F^2 \qquad (10)$$

where α is to balance the domain-invariance and category-discrimination of the learned features, and β is to avoid numerical instability issue. Clearly, to some extent, if α is small, the discriminative knowledge will be lost, and large α will impact the domain alignment. With respect to β, a small value may lead to a trivial solution, and a large value could affect the feature learning procedure. Thus, similar to [10], [26], we will choose reasonable values of α and β. Moreover, parameter sensitivity analysis is conducted in this article. The results demonstrate that DTLC could achieve comparable performance under a wide range of parameter values.

The overall discriminative transfer feature learning optimization of (10) with respect to $\mathbf{P}$ can be formulated as

$$\begin{aligned}
&\min_{\mathbf{P}} \ \mathrm{Tr}(\mathbf{P}^\top \mathbf{X} (\mathbf{W} + \alpha \mathbf{M}) \mathbf{X}^\top \mathbf{P}) + \beta \|\mathbf{P}\|_F^2 \\
&\text{s.t. } \mathbf{P}^\top \mathbf{X} \mathbf{H} \mathbf{X}^\top \mathbf{P} = \mathbf{I}_d \qquad (11)
\end{aligned}$$

where $\mathbf{I}_d$ is an identity matrix of dimension d and $\mathbf{H} = \mathbf{I}_{(n_s+n_t)} - (1/(n_s+n_t))\mathbf{1}_{(n_s+n_t)\times(n_s+n_t)}$ is the centering matrix. The constraint in (11) aims to maximize the variances for embedded source and target data, such as [10] and [26].

Clearly, the constrained nonlinear optimization problem (11) can be solved as a generalized eigendecomposition problem. The optimal solution of $\mathbf{P} \in \mathbb{R}^{m \times d}$ satisfying (11) will be obtained efficiently as

$$(\mathbf{X}(\mathbf{W} + \alpha \mathbf{M}) \mathbf{X}^\top + \beta \mathbf{I}_m) \mathbf{P} = \mathbf{X} \mathbf{H} \mathbf{X}^\top \mathbf{P} \Theta \qquad (12)$$

where $\Theta = \mathrm{diag}(\theta_1, \theta_2, \ldots, \theta_d) \in \mathbb{R}^{d \times d}$ is a diagonal matrix with Lagrange multipliers, and the generalized eigenvectors of (12) corresponding to the d-smallest eigenvalues are the optimal solution.

It is worth noting that the discriminative transfer feature learning of DTLC can be kernelized to be applied to nonlinear

scenarios. Consider the transformation ϕ as a kernel-mapping $\phi : \boldsymbol{x} \to \phi(\boldsymbol{x})$ to boost the adaptation power, and denote the kernel matrix for source and target data as $\mathbf{K} = \phi(\mathbf{X})^\top \phi(\mathbf{X}) \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$. By applying the representer theorem [46], the corresponding nonlinear formulation of (11) will be given as

$$\begin{aligned} &\min_{\mathbf{P}_k} \ \mathrm{Tr}\big(\mathbf{P}_k^\top \mathbf{K}(\mathbf{W} + \alpha\mathbf{M})\mathbf{K}^\top \mathbf{P}_k\big) + \beta\|\mathbf{P}_k\|_F^2 \\ &\text{s.t. } \mathbf{P}_k^\top \mathbf{K}\mathbf{H}\mathbf{K}^\top \mathbf{P}_k = \mathbf{I}_d. \end{aligned} \tag{13}$$

Similar to (11), (13) can also be addressed easily by employing the generalized eigendecomposition technique.

C. Label Consistency

Recently, many related works leverage target pseudo labels to align class-wise distributions of both domains [7], [10], [26], [28], and the correctness of target pseudo labels plays a vital role in the next round of distribution matching. However, most methods just infer target labels through source classifiers, and these source classifiers may prone to be overfitting when directly applied for target data due to the existing domain shift, even in the learned domain-invariant embedding feature spaces. Therefore, considering the label consistency between- and within-domain data in the pseudo label inference step, we propose an efficient label refinement technique to improve the target data prediction accuracy, resulting in a more accurate class-wise distribution alignment across source and target.

1) Label Consistency Between Source and Target Domains: In the transformed feature space, we believe that if a target sample is closer to the source domain, the source classifier would predict it more accurately. In other words, the labels of source data and target data that are much similar to the source should be consistent. Hence, we can assign larger weights to these target data, which are close to the source domain, and give smaller weights to these target samples that are far away from the source. The weights imply the consistency between each target sample and the source domain, as well as describe our confidence to the target pseudo labels.

Inspired by the idea of domain classifier [22], [47], a linear classifier to separate source and target, we can effectively measure the closeness of each target sample and the source domain by utilizing the Euclidian distance from this sample to the domain classifier. Based on the distance, all the weights for target data can be computed correspondingly.

Actually, in many scenarios, a simple linear function cannot distinguish source and target well. We tend to train multiple class-wise domain classifiers to classify each class of source domain and the whole target domain instead, since this manner could make the weights of target data more distinguishable. In the learned feature space, we denote the domain classifier of source data in class c and the target domain as $f_{dc}^{(c)}$, and the distance between the projected target data $z_{t_i}^{(c)}$ with pseudo label c and $f_{dc}^{(c)}$ as $l_{z_{t_i}^{(c)}}$. Then, we calculate the weight for $z_{t_i}^{(c)}$ using a simple monotone decreasing function as

$$\lambda_{z_{t_i}^{(c)}} = \frac{n_t}{n_t^{(c)}} \cdot \frac{1}{1 + \exp\left(\eta \cdot l_{z_{t_i}^{(c)}}\right)} \tag{14}$$

where η is a scale parameter, which could rescale the distance to a reasonable range. If the value of η is small, all the distances between target data and the domain classifier is undistinguishable. However, if η is large, the weights of target data will close to 0 or some constant, which definitely will impact the label propagation. A detailed parameter sensitivity analysis of η will be shown in Section IV, and the results verify that DTLC is not that sensitive to the value of η. $n_t/n_t^{(c)}$ is a coefficient to balance the effects of different classes as in [26]. Then, the formulation of label consistency between source and target can be formulated as

$$\min_{f_t} \ \sum_{i=1}^{n_t} \lambda_{z_{ti}} (f_s(z_{ti}) - f_t(z_{ti}))^2 \tag{15}$$

where $f_s(z_{ti})$ is the prediction of source classifier for z_{ti}, and $f_t(z_{ti})$ is the expected real target label. Obviously, (15) only has trivial solution, so we should consider the structural information of target domain.

2) Label Consistency Within Target Domain: To fully explore the geometrical information of target data in the embedded space, we expect that the labels of target data, which are close to each other, are consistent within the target domain. Resort to the manifold regularization theorem [48], we can introduce it into our label consistency loss function as

$$\sum_{i,j=1}^{n_t} b_{ij} (f_t(z_{ti}) - f_t(z_{tj}))^2 \tag{16}$$

where b_{ij} is used for measuring the similarity between z_{ti} and z_{tj}. We adopt binary weight strategy to define b_{ij} if z_{ti} is the p-nearest neighbor of point z_{tj}. Then, we can incorporate (15) and (16) into one loss function as

$$\begin{aligned} \min_{f_t} \ &\sum_{i=1}^{n_t} \lambda_{z_{ti}} (f_s(z_{ti}) - f_t(z_{ti}))^2 \\ &+\rho \sum_{i,j=1}^{n_t} b_{ij} \left(f_t(z_{ti}) - f_t(z_{tj})\right)^2 \end{aligned} \tag{17}$$

where ρ is a tradeoff parameter to balance the effects of label consistency between and within domains.

It is notable that the first term in (17) expects the final prediction $f_t(z_{ti})$ for z_{ti} to be close to $f_s(z_{ti})$ when weight $\lambda_{z_{ti}}$ is large. However, if $\lambda_{z_{ti}}$ is small, $f_t(z_{ti})$ could be different from $f_s(z_{ti})$ to some extent. The second term in (17) is a classical manifold regularization, which aims to explore the intrinsic geometry structure of target data. Moreover, introducing the manifold regularization enables DTLC propagating target labels from data with larger weights to data with small weights gradually.

3) Label Consistency Optimization: For simplicity, we define $\mathbf{Z}_t = [z_{t1}, \ldots, z_{tn_t}]$, $\Lambda = \mathrm{diag}(\lambda_{z_{t1}}, \ldots, \lambda_{z_{tn_t}})$, $\mathbf{F}_s = [f_s(z_{t1}), \ldots, f_s(z_{tn_t})]^\top$, and $\mathbf{F}_t = [f_t(z_{t1}), \ldots, f_t(z_{tn_t})]^\top$. $\mathcal{L}$ is the Laplacian matrix constructed within the target domain given by $\mathcal{L} = \mathbf{D} - \mathbf{B}$, where $\mathbf{B} = [b_{ij}]_{n_t \times n_t}$ and $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{n_t} b_{ij}$. Then, problem (17) can be rewritten as

$$\min_{\mathbf{F}_t} \ (\mathbf{F}_t - \mathbf{F}_s)^\top \Lambda (\mathbf{F}_t - \mathbf{F}_s) + \rho \mathbf{F}_t^\top \mathcal{L} \mathbf{F}_t. \tag{18}$$

We assume the expected target classifier to be $f_t(z) = \boldsymbol{w}^\top z$, and set the derivative of (18) with respect to $\boldsymbol{w}$ to zero. We can

Algorithm 1 DTLC Algorithm

Input:

Source data and labels: $\{\mathbf{X}_s, \boldsymbol{y}_s\}$; Target data: $\{\mathbf{X}_t\}$;
Subspace dimension: d; Iteration: T;
Tradeoff parameters: α, β, η.

Output: Target classifier f_t.

1: Construct MMD matrix $\mathbf{W}_0$ by Eq.(2);
2: Initialize $\mathbf{M}_0 = \mathbf{0}_{(n_s+n_t)\times(n_s+n_t)}$;
3: **repeat**
4: Obtain the projected matrix $\mathbf{P}$ by solving the generalized eigen-decomposition problem in Eq.(12)
5: $[\mathbf{Z}_s, \mathbf{Z}_t] = [\mathbf{P}^\top \mathbf{X}_s, \mathbf{P}^\top \mathbf{X}_t]$;
6: Use $\{\mathbf{Z}_s, \boldsymbol{y}_s\}$ to train a standard 1-NN source classifier f_s and predict target pseudo labels $\hat{\boldsymbol{y}}_t$;
7: Train C class-wise domain separators to calculate the weight matrix Λ by Eq. (15), and construct the graph laplacian $\mathcal{L}$ for target data;
8: Derive the refined target classifier f_t according to Eq.(19), and update target pseudo labels $\hat{\boldsymbol{y}}_t$;
9: Update $\mathbf{W}, \mathbf{M}$;
10: **until** Convergence or maximum iteration achieved.

obtain the optimal solution of (18) as

$$\boldsymbol{w}^* = \left(\mathbf{Z}_t(\Lambda + \rho\mathcal{L})\mathbf{Z}_t^\top\right)^{-1}\mathbf{Z}_t\Lambda\mathbf{F}_s. \tag{19}$$

Then, the refined label for $\boldsymbol{z}_{ti}$ can be derived as $f_t(\boldsymbol{z}_{ti}) = (\boldsymbol{w}^*)^\top \boldsymbol{z}_{ti}$ by label consistency.

Obviously, parameter ρ is very important to balance the two parts in (18). A large ρ value will emphasize the effect of label consistency within the target domain, whereas a small ρ value may not fully explore the knowledge of target geometrical structure. Once $f_t(\boldsymbol{z})$ is obtained, we can pick up ρ automatically from the sequence $[10^{(-5)}, 10^{(-4)}, \ldots, 10^4, 10^5]$ according to the performance of target classifier $f_t(\boldsymbol{z})$ on the weighted labeled source data as in [22].

Actually, the use of projected feature space not only eliminates the domain shift with discriminative information preserved but also helps us better utilize the label consistency of data between and within domains to refine target pseudo labels efficiently. The refined target labels will be provided to the next round of transfer feature learning and facilitate aligning class-wise distributions in a positive feedback way. Thus, the discriminative transfer feature learning and target label refinement will boost each other in a "win–win" feedback loop such that obtaining superior performance. To the best of our knowledge, DTLC is among the very leading approaches to incorporate discriminative transfer feature learning and coarse-to-fine label refinement into one framework.

D. DTLC Algorithm and Analysis

Based on the earlier discussion, we can summarize DTLC as in Algorithm 1.

1) Complexity Analysis: Here, we analyze the computational complexity of DTLC in Algorithm 1 using the big $\mathcal{O}$ notation. DTLC consists of two main parts: discriminative transfer feature learning and label consistency. Specifically, for discriminative transfer feature learning step, constructing MMD matrix $\mathbf{W}$ and distance matrix $\mathbf{M}$ costs around $\mathcal{O}(C(n_s + n_t)^2 + (n_s + n_t)^2)$, i.e., Lines 1 and 9; solving the generalized eigendecomposition problem costs $\mathcal{O}(dm^2)$, i.e., Line 4. For label consistency step, training C linear domain classifiers costs $\mathcal{O}(Cd(n_s + n_t))$, i.e., Line 7, and deriving the optimal f_t costs $\mathcal{O}(n_t^3)$. The remaining steps need around $\mathcal{O}(dm(n_s + n_t))$. In generalize, the overall computational complexity of DTLC is $\mathcal{O}(TC(n_s + n_t)^2 + Tdm(m + n_s + n_t) + Tn_t^3)$.

2) Transferability Analysis: To reduce the error rate on the unlabeled target data with labeled source data, the measurement of their discrepancy is decisive according to the domain adaptation theory [49]. In DTLC, we first explicitly minimize the MMD distances of both marginal and conditional distributions across domains and enhance the category discriminability of the learned representations. With the discrepancy measure, the generalization bound based on VC-dimension can be strictly derived for domain adaptation. Furthermore, we exploit the label consistency of source and target data from between- and within-domain perspectives to refine target pseudo labels, which improves generalization bound for crossed domain image classification. Noteworthily, the transfer feature learning and target label refinement are complementary to each other and are all crucial for learning ideal transferable models.

TABLE I
STATISTIC DESCRIPTIONS OF USED CROSS-DOMAIN DATA SETS

Dataset	#Feature	#Class	Domain	#Sample
Office-10 (SURF/DeCAF$_6$)	800/4,096	10	A, W, D	1,410
Caltech-10 (SURF/DeCAF$_6$)	800/4,096	10	C	1,123
Office-31 (DeCAF$_7$/ResNet-50)	4,096/2048	31	A, W, D	4,652
Office-Home (ResNet-50)	2048	65	Ar, Cl, Pr, Rw	15,500
ImageNet	4,096	5	ImageNet(I)	7,341
VOC2007	4,096	5	VOC2007(V)	3,376
CMU-PIE	1,024	68	C05,C07, C09, C27, C29	11,554

IV. EXPERIMENTS

In this section, we evaluate the performance of DTLC with massive SOTA shallow and deep domain adaptation (DA) methods on several popular visual cross-domain benchmarks. The code of DTLC is available online at https://github.com/hnjzlishuang/DTLC

A. Data Sets' Description

We adopt several popular cross-domain benchmarks, i.e., CMU-PIE, Office-31 (DeCAF$_7$, ResNet-50), ImageNet-VOC2007, Office+Caltech10 (SURF and DeCAF$_6$), and Office-Home (ResNet-50). Table I lists their statistics. In the following, we will introduce them in detail.

CMU-PIE [50] includes more than 40 000 face images of size 32 × 32 from 68 individuals. Considering the different pose factors, we focus on 5 out of 13 poses, such as in [10], i.e., C05 (left pose), C07 (upward pose), C09 (downward pose), C27 (front pose), and C29 (right pose). By randomly choosing two domains as source and target, we can obtain 20 cross-domain tasks, i.e., "C05→C07," "C05→C09," ...,"C29→C27."

ImageNet+VOC2007 is a large-scale standard data set [51], with images from common five classes (bird, cat, chair, dog, and person). We conduct two adaptation scenarios: ImageNet (I) → VOC2007 (V) and VOC2007 (V) → ImageNet (I).

Office-31 [52] is a standard benchmark for domain adaptation, which consists 4652 images with 31 categories collected from three distinct domains: Amazon (A) with images downloaded from the online web merchants, DSLR (D) with high-resolution images captured by a digital SLR camera, and Webcam (W) with low-resolution images recorded by a web camera. 4096-dim DeCAF$_7$ [53] and 2048-dim ResNet-50 [4] features are adopted to conduct six cross-domain tasks for evaluation, i.e., "A→D," "A→W," . . . , "W→D."

Caltech-256 (C) [54] is a widely used data set for object classification with 256 categories over 30 000 samples. We follow the standard protocols as [8] and select ten common object classes from Office31 and Caltech-256 to form Office+Caltech10: data set. Two kinds of features, 800-dim SURF and 4096-dim DeCAF$_6$ [53], are adopted; thus, we can thoroughly evaluate our approach by constructing 12 cross-domain tasks: "C→A," "C→W," . . . , "D→A" and "D→W."

Office-Home [55] consists of 65 different objects from four distinct domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw). For each domain, there are images found typically in Office and Home settings with different distributions. As [19], [37], 2048-dim ResNet-50 features are adopted for evaluation.

B. Baselines and Experimental Setup

To evaluate the effectiveness of our algorithm, we compare the performance of DTLC with massive SOTA shallow DA methods as follows: 1-nearest neighbor classifier (1NN) [56], principal component analysis (PCA) [57], GFK [8], TCA [9], joint distribution adaptation (JDA) [10], TJM [24], subspace alignment (SA) [58], label and structural consistency (LSC) [28], domain-invariant projection (DIP) [23], optimal transport for domain adaptation (OT-GL) [35], discriminative transfer subspace learning (DTSL) [33], RTML [14], DICD feature learning [26], joint geometrical and statistical alignment (JGSA) [7], and LDADA [36].

To further verify the superiority of DTLC with generic deep features, we also compare against several deep learning and unsupervised deep domain adaptation methods: AlexNet [51], ResNet [4], DDC [39], DAN [18], RTN [17], deep correlation alignment (DCORAL) [59], deep unsupervised convolutional domain adaptation (DUCDA) [60], domain adversarial neural networks (DANN) [30], adversarial discriminative domain adaptation (ADDA) [40], and maximum classifier discrepancy (MCD) [19].

Specifically, we rerun 1NN [56], PCA [57], GFK [8], TCA [9], JDA [10], TJM [24], DICD [26], JGSA [7], LDADA [36], and DTSL [33] using their released codes and tune parameters according to their default parameters. In addition, we report the results of other methods according to their original articles if the experimental settings are identical with ours.

In all our experiments, we set $\alpha = 1$ and $\eta = 1$ since they are not sensitive to the final performance. Besides, similar to [10] and [26], we set $\beta = 0.1$ for CMU-PIE data set and $\beta = 1$ for other data sets. We fix the maximum number of iterations as $T = 10$. Also, in Section IV-D7, we give parameter sensitivity analysis that will verify that for a wide range of parameter values, DTLC can always achieve stable performance.

For a fair comparison, similar to [7] and [26], we choose 1NN as the basic classifier for all other baselines since we do not need to tune the hyperparameters. Besides, in Section IV-D3, we will discuss DTLC with three different base classifiers, i.e., 1NN, SVM, and fully connected neural network (FC, i.e., softmax classifier). The results demonstrate that DTLC is not sensitive to the choice of different base classifiers. Top-1 classification accuracy for target data will be the evaluation metric.

C. Results and Discussion

In this section, we compare DTLC with extensive relevant domain adaptation methods for visual domain adaptation to demonstrate the effectiveness of our method.

1) Results on the CMU-PIE Data Set: We first carry out various experiments on CMU-PIE data set and the cross-domain facial image recognition task and compare DTLC with several popular shallow DA methods, including DICD [26], DTSL [33], LSC [28], and JGSA [7]. The comparison results shown in Table II indicate that DTLC obtains the best accuracies in all the tasks. To be specific, the average classification accuracy of DTLC in 20 tasks is 85.8%, which is more accurate than the best baseline DICD by a considerable margin.

It is worth noticing that DIP, DICD, and JGSA also adopt the class-discriminative information to boost their final performance. However, DTLC has incorporated both discriminative transfer feature learning and label consistency between and within domains into one framework. More than enforcing the embedded features to be distinguishable in terms of different classes, the label consistency of DTLC would further refine the target labels by considering the structural knowledge across domains, which is a great help for transferring knowledge from source to target effectively. Therefore, DTLC can generalize well on the unlabeled target domain.

From the results' comparison between JDA and LSC, and DICD and DTLC, we can notice that the performance of combining transfer feature learning with label consistency of source and target data is better than the standard domain-invariant feature extraction. This phenomenon demonstrates that the transfer feature learning and target label refinement should be coupled to benefit each other and are equally important for addressing visual domain adaptation problems.

Based on JDA, LSC explores label prorogation of target data to refine target pseudolabels and achieves improvement to some extent. This could attribute to the wrongly predicted target pseudolabels, which will hinder the correctness of label propagation process. Different from LSC, DTLC will trust the predictions of target data, which are closer to the source domain, with high confidence. The target pseudolabels can be updated according to the label consistency effectively.

Finally, as can be seen from Table II that the results of no adaptation methods perform poorly in many tasks, which reveals that the adaptation difficulty varies a lot in these 20 tasks, while DTLC could perform stably and achieve many SOTA results, implying the robustness and effectiveness of DTLC in face recognition domain adaptation problems.

2) Results on the ImageNet+VOC2007 Data Set: Here, we evaluate the performance of DTLC with relatively big data set. We directly exploit all the source and target samples for the experiment. The results are shown in Table III. DTLC

TABLE II
ACCURACY (%) ON CMU-PIE DATA SET

Task\Method	1NN	PCA	GFK	TCA	JDA	SA	LSC	DIP	OT-GL	DTSL	RTML	DICD	JGSA	LDADA	**DTLC**
C05→C07	26.1	24.8	26.2	40.8	58.6	26.8	59.0	29.9	59.4	65.9	60.1	73.0	52.9	34.5	**85.1**
C05→C09	26.6	25.2	27.3	41.8	52.0	28.2	52.1	32.8	58.7	64.1	55.2	72.0	53.1	44.9	**82.7**
C05→C27	30.7	29.3	31.2	59.6	83.7	30.9	84.0	36.7	-	82.0	85.2	92.2	66.0	61.5	**97.1**
C05→C29	16.7	16.3	17.6	29.4	47.7	19.6	48.2	12.7	48.4	54.9	53.0	66.9	46.1	35.4	**77.2**
C07→C05	24.5	24.2	25.2	41.8	60.6	26.4	61.0	25.8	61.9	45.0	58.1	69.9	57.5	31.4	**82.8**
C07→C09	46.6	45.5	47.4	51.5	60.2	48.0	60.5	53.4	64.4	53.5	63.9	65.9	57.2	34.9	**83.9**
C07→C27	54.1	53.4	54.3	64.7	75.4	54.3	76.2	50.1	-	71.4	76.2	85.3	69.2	53.5	**92.1**
C07→C29	26.5	25.4	27.1	33.7	40.9	28.2	41.4	29.5	52.7	48.0	40.4	48.7	49.8	26.4	**79.7**
C09→C05	21.4	21.0	21.8	34.7	50.9	23.2	51.4	22.7	57.9	52.5	53.1	69.4	56.1	38.2	**80.0**
C09→C07	41.0	40.5	43.2	47.7	56.1	44.3	56.7	36.3	64.7	55.6	58.7	65.4	58.7	30.5	**84.4**
C09→C27	46.5	46.1	46.4	56.2	68.0	46.2	69.5	45.8	-	77.5	69.8	83.4	69.5	60.6	**94.3**
C09→C29	26.2	25.3	26.7	33.2	40.3	28.9	40.6	20.2	52.8	54.1	42.1	61.4	52.2	40.7	**79.9**
C27→C05	33.0	32.0	34.2	55.6	81.0	36.3	81.5	31.4	-	81.5	81.1	93.1	63.2	61.3	**96.7**
C27→C07	62.7	61.0	62.9	67.8	82.8	63.8	83.1	67.5	-	85.4	83.9	90.1	65.7	56.7	**94.8**
C27→C09	73.2	72.2	73.4	75.9	87.2	73.2	87.2	76.8	-	82.2	89.5	89.0	62.6	67.8	**95.4**
C27→C29	37.2	35.1	37.4	40.3	49.9	38.1	50.5	36.5	-	72.6	56.3	75.6	57.0	50.4	**84.4**
C29→C05	18.5	18.9	20.4	27.0	47.5	23.4	47.8	14.2	45.7	52.2	29.1	62.9	56.3	31.3	**75.4**
C29→C07	24.2	23.4	24.6	30.0	44.8	25.5	45.5	29.3	51.3	49.4	33.3	57.0	54.7	24.1	**77.8**
C29→C09	28.3	27.2	28.5	30.0	48.1	28.6	49.1	31.7	52.6	58.5	39.9	65.9	56.4	35.4	**82.4**
C29→C27	31.2	30.3	31.3	33.6	56.5	31.2	57.3	26.3	-	64.3	47.1	74.8	61.7	48.2	**89.7**
Average	34.8	33.9	35.4	44.8	59.6	36.3	60.1	35.5	-	63.5	58.8	73.1	58.3	43.4	**85.8**

TABLE III
ACCURACY (%) ON IMAGENET + VOC2007 DATA SET

Task/Method	1NN	JDA	TJM	LSC	DTSL	DICD	JGSA	LDADA	**DTLC**
I→V	50.8	63.4	63.7	**67.2**	62.0	60.2	52.3	65.1	64.8
V→I	38.2	70.2	73.0	77.2	73.8	65.8	70.6	53.4	**85.8**
Average	44.5	66.8	68.4	72.2	67.9	63.0	61.5	59.3	**75.3**

TABLE IV
ACCURACY (%) ON OFFICE-31 DATA SET WITH DECAF$_7$ FEATURES COMPARED WITH SHALLOW DA METHODS

Task/Method	1NN	JDA	TJM	SA	LSC	DIP	DTSL	DICD	JGSA	LDADA	**DTLC**
A→D	59.6	56.8	58.4	61.0	63.5	56.0	60.0	62.7	63.5	68.0	**68.4**
A→W	54.0	58.1	51.7	59.5	60.0	51.9	54.5	56.7	56.0	63.0	**66.1**
D→A	42.4	44.8	43.3	46.9	47.7	44.0	46.6	54.2	58.5	56.2	**59.0**
D→W	90.9	95.7	92.8	95.1	96.0	95.3	94.3	96.5	97.2	83.7	**97.4**
W→A	40.8	46.2	41.9	46.6	49.5	42.3	45.6	51.2	55.4	55.0	**56.5**
W→D	97.8	97.6	97.4	98.2	97.8	**98.8**	94.0	98.6	98.2	93.6	**98.8**
Average	64.3	66.5	64.3	67.9	69.1	64.7	65.8	70.0	71.5	69.9	**74.4**

generally exceeds other methods in terms of average accuracy, which is 75.3%, gaining a significant performance improvement of 3.1% compared to the best baseline LSC. On the first task I→V, LSC obtains the highest accuracy. However, on task V→I, DTLC has a significant advantage, which improves 8.6% over LSC. On the one hand, both methods exploit the target pseudolabels for adaption; thus, both of them achieve better performance. On the other hand, LSC adopts all of the target pseudolabels in order to propagation. In this way, the accuracy of the predicted labels will directly affect the final recognition. Different from LSC, we merely select the most reliable target samples and use them for label propagation.

3) Results on the Office-31 Data Set: Table IV illustrates the classification accuracies of shallow domain adaptations on Office-31 with DeCAF$_7$ features. We compare DTLC with several traditional methods, including the latest LDADA [36] and JGSA [7], to verify the effectiveness of DTLC on this more complex data set. It is worth noting that DTLC achieves much better performance than all the baseline methods on all tasks.

TABLE V
ACCURACY (%) ON OFFICE-31 DATA SET WITH DECAF$_7$ AND RESNET-50 FEATURES COMPARED WITH DEEP DA METHODS

Features	Method/Task	A→D	A→W	D→A	D→W	W→A	W→D	Average
DeCAF$_7$	AlexNet	63.8	61.6	51.1	95.4	49.8	99.0	70.1
	DDC	64.4	61.8	52.1	95.0	52.2	98.5	70.7
	DAN	67.0	68.5	54.0	96.0	53.1	99.0	72.9
	RTN	**71.0**	**73.3**	50.5	96.8	51.0	99.6	73.7
	DCORAL	66.4	66.8	52.8	95.7	51.5	99.2	72.1
	DUCDA	68.3	68.3	53.6	96.2	51.6	**99.7**	73.0
	DTLC	68.4	66.1	**59.0**	**97.4**	**56.5**	98.8	**74.4**
ResNet-50	ResNet	68.9	68.4	62.5	96.7	60.7	99.3	76.1
	DAN	78.6	80.5	63.6	97.1	62.8	99.6	80.4
	RTN	77.5	84.5	66.2	96.8	64.8	99.4	81.6
	JAN	84.7	85.4	68.6	97.4	70.0	99.8	84.3
	DANN	79.7	82.0	68.2	96.9	67.4	99.1	82.2
	ADDA	77.8	86.2	69.5	96.2	68.9	98.4	82.9
	MADA	87.8	**90.0**	70.3	97.4	66.4	99.6	85.2
	MCD	**92.2**	88.6	69.5	**98.5**	69.7	**100.0**	86.5
	DTLC	91.0	85.3	**73.5**	98.0	**73.5**	99.8	**86.9**

TABLE VI
ACCURACY (%) ON OFFICE+CALTECH10 DATA SET WITH DECAF$_6$ FEATURES COMPARED WITH DEEP DA METHODS

Task/Method	AlexNet	DDC	DAN	RTN	DCORAL	**DTLC**
C→A	91.9	91.9	92.0	**93.7**	92.4	92.8
C→W	83.7	85.4	90.6	96.9	91.1	**98.0**
C→D	87.1	88.8	89.3	**94.2**	91.4	93.0
A→C	83.0	85.0	84.1	88.1	84.7	**88.2**
A→W	79.5	86.1	91.8	**95.2**	-	93.6
A→D	87.4	89.0	91.7	**95.5**	-	87.3
W→C	73.0	78.0	81.2	86.6	79.3	**88.1**
W→A	83.8	84.9	92.1	**92.5**	-	92.0
W→D	**100.0**	**100.0**	**100.0**	**100.0**	-	**100.0**
D→C	79.0	81.1	80.3	84.6	82.8	**89.3**
D→A	87.1	89.5	90.0	**93.8**	-	92.9
D→W	97.7	98.2	98.5	99.2	-	**100.0**
Average	86.1	88.2	90.1	**93.4**	-	93.0

Moreover, we also show several recent end-to-end deep domain adaptation methods for comparison, and Table V summarizes the comparison results. DTLC obtains the highest accuracies on two hard tasks: D→A and W→A with both DeCAF$_7$ and ResNet-50 features. Although these MMD- and adversarial-based deep methods are able to achieve relatively

TABLE VII
ACCURACY (%) ON OFFICE+CALTECH10 DATA SET WITH DECAF$_6$ AND SURF FEATURES COMPARED WITH SHALLOW DA METHODS

Task/Method	1NN	PCA	GFK	TCA	JDA	TJM	SA	LSC	DIP	OT-GL	DTSL	RTML	DICD	JGSA	LDADA	DTLC
C→A	87.3	88.1	88.2	89.8	89.8	88.8	80.8	91.3	78.9	92.2	91.5	90.6	91.0	91.4	92.5	**92.8**
C→W	72.5	83.4	77.6	78.3	83.7	81.4	86.0	85.1	80.9	83.8	76.6	85.4	92.2	86.8	86.4	**98.0**
C→D	79.6	84.1	86.6	85.4	86.6	84.7	76.3	91.7	69.8	85.4	87.9	89.3	**93.6**	**93.6**	88.0	93.0
A→C	71.7	79.3	79.2	82.6	82.3	84.3	89.4	85.1	**89.8**	87.2	85.8	86.4	86.0	84.9	88.6	88.2
A→W	68.1	70.9	70.9	74.2	78.6	71.9	90.4	79.7	84.7	84.5	73.6	80.3	81.4	81.0	90.5	**93.6**
A→D	74.5	82.2	82.2	81.5	80.3	76.4	80.0	79.0	72.2	85.3	82.2	84.4	83.4	**88.5**	85.0	87.3
W→C	55.3	70.3	69.8	80.4	83.5	83.0	87.1	85.0	85.3	83.7	72.8	83.3	84.0	85.0	87.0	**88.1**
W→A	62.6	73.5	76.8	84.1	90.2	87.6	81.4	89.5	75.5	**92.0**	75.5	91.4	89.7	90.7	**92.0**	**92.0**
W→D	98.1	99.4	**100.0**	**100.0**	**100.0**	**100.0**	98.3	**100.0**	98.6	91.4	**100.0**	**100.0**	**100.0**	**100.0**	96.8	**100.0**
D→C	42.1	71.7	71.4	82.3	85.1	83.8	83.7	85.3	72.4	84.9	75.2	85.7	86.1	86.2	86.6	**89.3**
D→A	50.0	79.2	76.3	89.1	91.4	90.3	79.3	91.5	70.3	**92.9**	85.0	91.9	92.1	92.0	90.9	**92.9**
D→W	91.5	98.0	99.3	99.7	99.0	99.3	98.7	**100.0**	99.4	94.2	99.3	99.0	99.0	99.7	95.0	**100.0**
Average	71.1	81.7	81.5	85.6	87.5	86.0	86.0	81.5	88.2	87.6	83.8	89.0	89.9	90.0	89.9	**93.0**
DeCAF$_6$⇑ and SURF⇓ features																
Task/Method	1NN	PCA	GFK	TCA	JDA	TJM	SA	LSC	DIP	OT-GL	DTSL	RTML	DICD	JGSA	LDADA	DTLC
C→A	23.7	39.5	46.0	45.6	43.1	46.8	39.8	45.6	40.0	44.2	51.3	49.3	47.3	51.5	**54.8**	50.3
C→W	25.8	34.6	37.0	39.3	39.3	39.0	36.9	40.3	36.9	38.9	38.7	44.7	46.4	45.4	**60.2**	54.4
C→D	25.5	44.6	40.8	45.9	49.0	44.6	37.6	47.8	35.9	44.5	47.1	47.6	49.7	45.9	41.5	**52.4**
A→C	26.0	39.0	40.7	42.0	40.9	39.5	42.1	41.2	40.8	34.6	43.4	43.7	42.4	41.5	38.4	**46.6**
A→W	29.8	35.9	37.0	40.0	38.0	42.0	45.9	38.3	45.2	37.0	36.6	44.3	45.1	45.8	**49.3**	48.1
A→D	25.5	33.8	40.1	35.7	42.0	45.2	32.2	38.2	37.3	38.9	38.8	43.9	38.9	**47.1**	39.1	45.4
W→C	19.9	28.2	24.8	31.5	33.0	30.2	34.2	30.6	31.5	**36.0**	29.8	34.8	33.6	33.2	31.7	33.8
W→A	23.0	29.1	27.6	30.5	29.8	30.0	32.5	27.9	30.6	39.4	34.1	35.3	34.1	**39.9**	35.1	33.5
W→D	59.2	89.2	85.4	91.1	**92.4**	89.2	88.5	89.2	83.7	84.0	82.8	91.0	89.8	90.5	74.6	87.3
D→C	26.3	29.7	29.3	33.0	31.2	31.4	34.3	30.8	27.6	32.4	30.1	**34.6**	**34.6**	29.9	29.9	32.1
D→A	28.5	33.2	28.7	32.8	33.4	32.8	28.8	32.1	28.8	37.2	32.1	33.3	34.5	38.0	**40.6**	36.2
D→W	63.4	86.1	80.3	87.5	89.2	85.4	88.5	90.9	91.7	81.1	72.2	89.0	91.2	91.9	74.7	**92.9**
Average	31.4	43.6	43.1	46.2	46.8	46.3	45.1	46.1	44.2	47.7	45.7	49.3	49.0	50.1	47.5	**51.1**

good recognition performance, the time complexity is still a big challenge for deep methods. Carefully observing the average accuracy, our method is marginally better than these deep-learning-based approaches, which verifies the superiority of DTLC when equipped with deep abstract features.

4) Results on the Office-Caltech10 Data Set: First, we perform comparative experiments with high-dimensional DeCAF$_6$ deep features. As shown in Table VII, DTLC easily beats the shallow methods and obtains the best average accuracy winning 9 out of 12 tasks, and the second-best approach JGSA wins 3 out of 12 cross-domain pairs. For the average accuracy, DTLC also leads JGSA by 3.0%. Compared with deep domain adaptation methods in Table VI, DDC, DAN, and so on, DTLC still wins 6 out of 12 tasks and leads the average accuracy by 4.8% and 2.9% in terms of the average accuracy, respectively. RTN is a SOTA deep DA method based on AlexNet winning 7 out of 12 tasks with marginal improvement than DTLC. However, for the other five tasks, DTLC is better than RTN with large margins. In summary, based on the general deep features, the performance of DTLC is comparable to or even better than several competitive deep domain adaptation methods.

Second, the results of SURF features are shown in Table VII. DICD and JGSA outperform other shallow baselines, e.g., JDA, on most tasks because they all preserve discriminative knowledge when aligning source and target domains. However, they still perform inferiorly to DTLC because of that they learn feature embeddings and target pseudolabels separately. These two procedures cannot benefit from each other. DTLC explicitly refines target pseudolabels by considering label consistency in an effective way, which would guide the next round of classwise distributions alignment. This positive feedback loop could significantly improve the classification accuracy.

Finally, it is observed that the results on DeCAF$_6$ are much better than that on SURF, suggesting that the deep representations can bridge the domain shift to some extent.

TABLE VIII
ACCURACY (%) ON OFFICE-HOME DATA SET WITH RESNET-50 FEATURES COMPARED WITH DEEP DA METHODS

Task/Method	ResNet	DAN	JAN	DANN	MCD	DTLC
Ar→Cl	34.9	43.6	45.9	45.6	46.9	**56.0**
Ar→Pr	50.0	57.0	61.2	59.3	64.1	**74.9**
Ar→Rw	58.0	67.9	68.9	70.1	**77.6**	76.5
Cl→Ar	37.4	45.8	50.4	47.0	56.1	**57.4**
Cl→Pr	41.9	56.5	59.7	58.5	62.4	**71.3**
Cl→Rw	46.2	60.4	61.0	60.9	65.5	**70.9**
Pr→Ar	38.5	44.0	45.8	46.1	58.9	**61.0**
Pr→Cl	31.2	43.6	43.4	43.7	45.8	**52.4**
Pr→Rw	60.4	67.7	70.3	68.5	**80.0**	76.3
Rw→Ar	53.9	63.1	63.9	63.2	**73.3**	68.3
Rw→Cl	41.2	51.5	52.4	51.8	49.8	**56.7**
Rw→Pr	59.9	74.3	76.8	76.8	**83.1**	80.4
Average	46.1	56.3	58.3	57.6	63.6	**66.8**

5) Results on the Office-Home Data Set: Table VIII summarizes the results of recent deep methods on Office-Home with ResNet-50 features. We compare DTLC with latest MCD [19] and DANN [30] to verify the effectiveness of DTLC on this more complex data set. It is worth noting that DTLC achieves the best adaptation performance on almost all tasks. Recent works also reveal that deep discriminative features

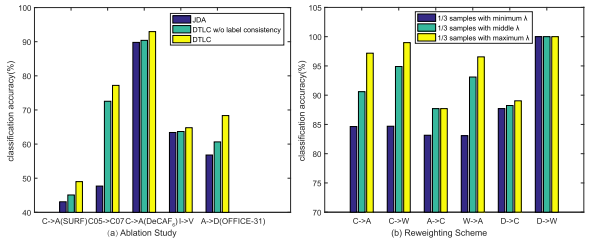


Fig. 3. (a) Results of JDA, DTLC without label consistency, and DTLC. (b) Performance comparison of every $1/3$ target data with different weight value λ from small to large. There are three bars in each task representing the classification accuracy of target data with $1/3$ minimum, middle, and maximum λ predicted by the source classifier.

TABLE IX

STATISTICS SUMMARIZATION. CASE 1: HOW MANY SAMPLES ARE CORRECTLY PREDICTED BY DTLC WITHOUT LABEL CONSISTENCY BUT WRONGLY PREDICTED BY DTLC. CASE 2: VICE VERSA. DATA SET: OFFICE-31. LC = LABEL CONSISTENCY

	A→D	A→W	D→A	D→W	W→A	W→D
DTLC w/o LC √ DTLC ×	24	36	111	5	112	1
DTLC w/o LC × DTLC √	65	120	363	16	295	5

can not only reduce but also not remove the cross-domain discrepancy [18]. Hence, equipping DTLC with deep generic features could further mitigate the domain shift and efficiently achieve attractive results compared with deep methods.

D. Empirical Analysis

1) Ablation Study: To deeply understand the contributions of different parts in DTLC, we conduct ablation study for DTLC. We evaluate over JDA, DTLC without (w/o) label consistency, and DTLC by randomly selecting one task from each benchmark and report their results in Fig. 3(a).

From the results, we can clearly observe that the preserved discriminative information could generally improve the performance over domain-invariant feature learning, i.e., DTLC without label consistency is invariably better than JDA. There is no doubt that adding class discriminative information to transfer feature learning could achieve a more effective discriminative model and a boosting of target classification accuracy. Moreover, when we incorporate transfer feature learning and label consistency into one framework, DTLC can further increase target classification accuracy by 3%–10% compared to DTLC without label consistency. This implies that target pseudolabel refinement is of vital important to learn ideal transferable features. We can conclude that both of them are indispensable and complement to each other.

To statistically verify the importance of label consistency, we count up how many samples are correctly predicted by DTLC but wrongly predicted by DTLC without label consistency, and vice versa for Office-31 data set. The results are listed in Table IX. Besides, in order to intuitionally show the effectiveness of label consistency, we randomly select some samples in Case 2 of task D→A in Table IX, as shown in Fig. 4. It is interesting to observe that the label consistency could help DTLC further predict plenty of target data correctly but little predicted wrongly, which effectively verifies the correctness of our hypothesis and implies that label consistency is crucial for domain adaptation.

Fig. 4. Randomly select some samples of task D→A (Case 2) in Table IX.

TABLE X

ACCURACY (%) OF VARIANTS OF BASE CLASSIFIER

Task/Method	1NN	DTLC_{1NN}	SVM	DTLC_{SVM}	FC	DTLC_{FC}
A→D	79.1	91.0	81.3	**91.4**	82.1	88.8
A→W	75.8	85.3	74.6	84.3	75.2	**88.1**
D→A	60.2	**73.5**	64.4	72.1	64.2	72.3
D→W	96.0	**98.0**	96.4	97.6	91.7	95.7
W→A	59.9	73.5	64.7	**73.8**	61.3	74.1
W→D	99.4	**99.8**	99.4	99.2	98.2	98.4
Average	78.4	**86.9**	80.1	86.4	78.8	86.2
Office-31⇑ and **Office-Home** ⇓ features						
Task/Method	1NN	DTLC_{1NN}	SVM	DTLC_{SVM}	FC	DTLC_{FC}
Ar→Cl	45.3	56.0	48.0	56.5	44.0	**56.6**
Ar→Pr	60.1	74.9	67.0	**76.6**	59.9	75.3
Ar→Rw	65.8	76.5	73.6	78.4	67.0	**78.6**
Cl→Ar	45.7	57.4	46.7	57.6	45.4	**57.9**
Cl→Pr	57.0	**71.3**	60.7	**71.3**	57.6	70.5
Cl→Rw	58.7	70.9	62.6	71.4	58.8	**71.8**
Pr→Ar	48.1	**61.0**	49.1	58.0	44.0	60.1
Pr→Cl	42.9	**52.4**	41.3	50.7	41.4	51.4
Pr→Rw	68.9	76.3	71.0	**76.8**	69.3	76.4
Rw→Ar	60.8	**68.3**	60.9	66.5	60.6	67.5
Rw→Cl	48.3	**56.7**	48.5	55.9	47.4	55.7
Rw→Pr	74.7	**80.4**	76.9	79.3	75.8	79.7
Average	56.4	**66.8**	58.9	66.6	55.9	**66.8**

2) Verification of Reweighting Scheme in Label Consistency: In label consistency step, we assign these target data that are closer to the source domain larger weights. We assume that the source classifier will predict target data with larger weights more accurately, and Fig. 3(b) verifies this assumption. We divide the target data into three folds according to their weights from small to large and leverage the source classifier to predict them. Six tasks of Office+Caltech10 (DeCAF_6) are randomly selected.[1] From the results, all tasks have the same phenomenon that the target data with larger weights have better prediction results by the source classifier. This verifies the effectiveness of our reweighting scheme in label consistency.

3) Variants of Base Classifier: To discuss the effects of different base classifiers selection, we, respectively, adopt 1NN, support vector machine (SVM), and fully connected neural network (FC, i.e., softmax classifier) as the base classifier. Table X lists the results of DTLC with each of these classifiers. DTLC_{1NN}, DTLC_{SVM}, and DTLC_{FC} represent 1NN, SVM, and FC and are chosen as the base classifiers, respectively. From the results, we could observe that each of them can achieve comparable performance. There is no doubt that DTLC is not that sensitive to the choice of different base classifiers.

4) Variants of DTLC: To extensively study the effectiveness of DTLC from different perspectives, we propose two variants of DTLC.

1) *JDA + Label Consistency (JDA-LC)* By incorporating the label consistency into JDA.

[1]We do not show the results of all the tasks due to space constraints, and other tasks also have the same trend.

TABLE XI
ACCURACY (%) OF DTLC VARIANTS

Dataset	Office+Caltech10 (SURF)		Office+Caltech10 (DeCAF$_6$)		CMU-PIE		ImageNet+VOC2007		OFFICE-31 (DeCAF$_7$)	
Method/Task	C → D	A → C	C → D	A → C	C05→C27	C05→C29	I → V	V → I	A → D	W → A
JDA-LC	50.5	43.6	87.1	91.9	88.5	63.2	64.4	85.8	97.0	52.8
DTLC-random	51.2	44.2	89.8	92.7	91.1	67.4	64.5	**85.9**	97.0	52.0
DTLC	**54.4**	**46.6**	**92.2**	**92.8**	**94.3**	**75.4**	**64.8**	85.8	**97.4**	**59.0**

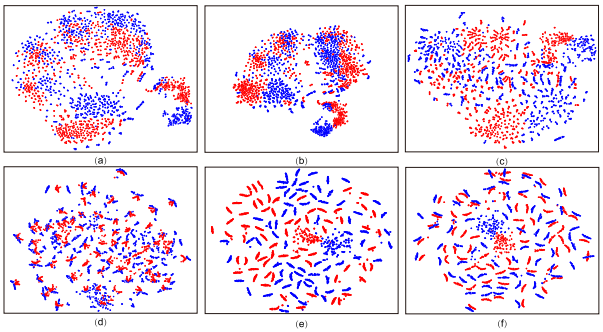


Fig. 5. T-SNE visualization of task C07 → C29 in CMU-PIE. Source and target samples are red and blue dots, respectively. Each cluster is corresponding to each class data predicted by corresponding method. (a) Original data. (b) TCA. (c) JDA. (d) JGSA. (e) DICD. (f) DTLC (Ours).

2) *DTLC-Random* By randomly choosing one positive data pair and one negative data pair for each data to penalize and other procedures are consistent with DTLC.

Table XI shows the results of JDA-LC, DTLC-random, and DTLC on randomly choosing tasks. From Table XI, JDA-LC performs worse than the other two on all the tasks, which indicates the vital importance of utilizing class discriminative information. Then, we compare DTLC with DTLC-random. From the results, we could observe that DTLC performs better than DTLC-random on almost all tasks, which verifies that discovering and optimizing the hardest data pairs in both domains could effectively mine category-discriminative information preserved in the learned feature space, which further compensates for the side effect of feature distortion and contributes to the accurate target classification.

5) Feature Visualization: First, in Fig. 5(a)–(f), we visualize the feature embeddings using t-SNE visualization [61] for Original data, TCA, JDA, JGSA, DICD and DTLC for task C07 → C29 of CMU-PIE, respectively. From Fig. 5, we can notice that original data are disorganized, and the data in the source and target domains follow diverse but related distributions. We could observe that TCA has a tendency to align the source and target domains. However, 65 clusters of TCA feature become obscure in the embedding space, which results from the mismatched classwise distributions alignment. This could explain the inferior performance of TCA. JDA aims to match both marginal and conditional distributions across domains, but the distorted features, as well as the wrongly-predicted target pseudolabels, would mix up different categories. Thus, most categories of JDA are poorly aligned. JGSA achieves much better alignment than JDA due to the source discriminative information preservation, while the target variance maximization may degrade its final performance. Compared with JGSA, DICD explicitly minimizes the intra-class compactness and maximizes the interclass dispersion for

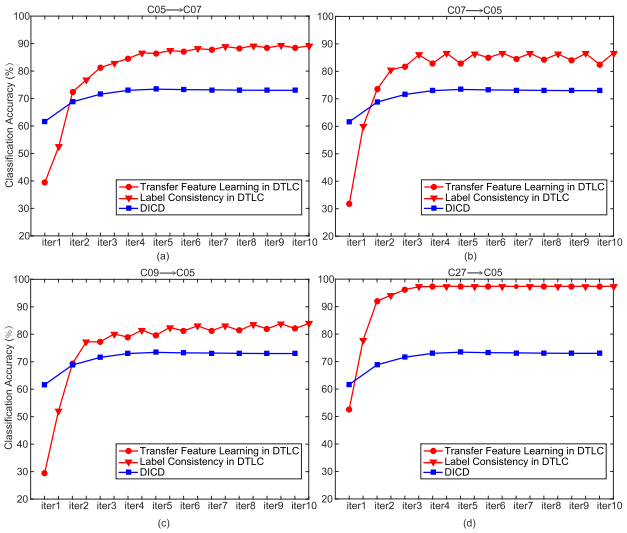


Fig. 6. Illustration of the iterative optimization process in DTLC for ten iterations. (a) C05 → C07. (b) C07 → C05. (c) C09 → C05. (d) C27 → C05.

source and target data, respectively. As displayed in Fig. 5(e), every cluster is distinguishable to others but with mismatched classwise alignment for source and target. This may attribute to the side effects of inaccurate target pseudolabels.

Different from all these methods, source and target data in different classes are perfectly matched in Fig. 5(f) since DTLC not only enforces the learned representations to be domain invariant and category discriminative but also refines target pseudolabels, leveraging the label consistency between and within domains. The iterative optimization strategy allows us to achieve more accurate classwise alignment with discriminative information preserved.

6) Iterative Optimization Study: In DTLC, we believe that the discriminative transfer feature learning can effectively mitigate the domain shift and allows us to better discover the shared knowledge between source and target domains. Moreover, the target pseudolabel refinement by label consistency can polish all the target labels and facilitate the discriminative transfer feature learning in return. DTLC has coupled these two procedures together to iteratively boost each other.

Fig. 6 represents the classification accuracy of DTLC during ten iterations and illustrates the importance of unifying the transfer feature learning and label consistency into one framework. We randomly choose several different domains of CMU-PIE to build four tasks in this experiment. The red circle represents the corresponding accuracy after transfer feature learning, and the red triangle represents the corresponding accuracy after label consistency in each iteration. We compare this with the best baseline DICD. The most difference between DTLC and DICD is that DTLC not only learns transfer feature

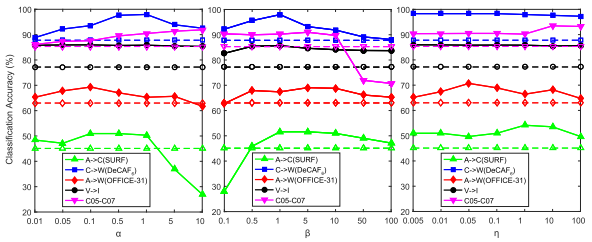


Fig. 7. Parameter sensitivity studies with respect to α, β, and η, respectively. The dashed lines show the best baseline results in the corresponding task.

but also explores label consistency, which emphasizes the importance of utilizing label consistency.

In each iteration, we merge transfer feature learning and label consistency into one framework. Before DTLC features reach a stable condition, each step, whether it is transfer feature learning or label consistency, could improve the target classification accuracy on the basis of the previous step. This implies that transfer features learning and label consistency are complementary to each other. Label consistency can further correct a part of the wrongly predicted target labels to benefit matching the classwise domain distributions more accurately. More accurate transfer feature learning can preserve necessary information of structure feature, which, in turn, boosts the effectiveness of label consistency. Both transfer feature learning and label consistency are optimized iteratively in our DTLC framework. The results after stabilization are much better than baselines, which demonstrates the superior of DTLC.

7) Parameter Sensitivity: We conduct parameter sensitivity studies with respect to α, β, and η in Fig. 7 to verify that under a wide range of parameter choices, DTLC could achieve desirable performance and outperform the best baseline. The results on A→C (SURF), C→W (DeCAF$_6$), A→W (OFFICE-31), V→I, and C05→C07 tasks are reported, while similar trends for all other tasks are not shown due to space limitation.

From Fig. 7, we can see that DTLC could achieve much better performance than baselines under a wide range of parameter values. We first run DTLC as α varies from 0.01 to 10, with β and η fixed. It can be observed that between 0.05 and 1, the classification accuracy grows with α, which indicates that learned features should be not only domain invariant but also category discriminative, while larger values of α could reduce the relative impact of domain alignment, which degrades the classification performance. Thus, we can choose $\alpha \in [0.01, 1]$.

We then evaluate DTLC by fixing α and η and varying the regularization parameter β from 0.1 to 100. In principle, smaller values of β will lead to numerical instability issues, which results in a decrease in target classification accuracy, and larger values of β could prevent DTLC learning effective domain transfer features. The bell-shaped curves in Fig. 7 also verify our analysis. Thus, we can choose $\beta \in [0.1, 10]$.

At last, we run DTLC with various values of η. We could observe that under a wide range of η choices, DTLC could achieve desirable performance and outperform the best baseline much better. The reason is that η is a scale factor of the distance between the source and target domains to the domainwise classification surface. Regardless of the value of η, the relative size of the distance does not change. Therefore, η could vary widely, which has less impact on the classification accuracy. Thus, we can choose $\eta \in [0.005, 100]$.

We can notice that DTLC could achieve much better performance than baselines under a wide range of parameter values. Note that in practice, DTLC is not that sensitive to η, and we can select $\eta \in [0.005, 100]$ first. Then, if we choose reasonable $\alpha \in [0.01, 1]$ and $\beta \in [0.1, 10]$, DTLC could perform stably and outperform other methods, which claims the robustness and effectiveness of DTLC. In this article, we just set $\alpha = 1$ and $\eta = 1$ as default throughout the experiments for easy implement.

TABLE XII

AVERAGES AND STANDARD ERRORS OF CLASSIFICATION ACCURACY (%) ON OFFICE-31 DATA SET WITH SURF FEATURES

Task/Method	NN	TCA	JDA	RTML	DICD	JGSA	LDADA	DTLC
A→D	35.5±2.7	47.3±2.8	49.0±2.6	48.9±2.6	54.0±1.5	58.5±2.5	53.8±1.9	**61.6±2.1**
D→W	37.0±1.7	63.4±1.9	68.9±1.5	69.3±1.5	74.4±2.4	75.7±1.9	63.9±2.3	**77.5±1.8**
W→D	47.7±2.3	61.4±2.8	65.4±1.9	71.7±2.2	74.1±2.4	75.2±2.1	61.3±1.7	**77.3±2.3**

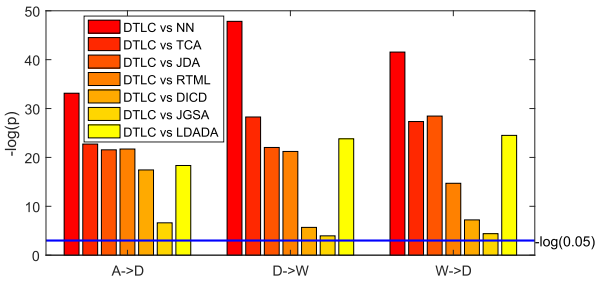


Fig. 8. p-value of the significance test (t-test) for results of DTLC versus other methods on Office-31.

8) Statistical Analysis: To further prove the superiority of DTLC, as introduced in Section IV-A, Office-31 is a very complex data set with 31 classes. Following the standard protocol in [6] and [26], we adopt 800-dim SURF features and randomly select 20 labeled samples for Amazon and eight for DSLR or Webcam for each class. Moreover, for the target domain, we randomly select three labeled samples as training data and the rest as unlabeled testing data. The averages and standard errors of classification accuracy are shown in Table XII.

To illustrate statistical significance more clearly, we conduct a significance test (t-test) in Fig. 8. Here, the significance level of 0.05 is utilized, and if the $p-$value is smaller than 0.05, the performance of the two methods is statistically significant. The blue line indicates the base level of 0.05 ($-log(0.05)$) with regard to each task. The larger the $-log(p)$, the greater the importance of DTLC compared to others. Based on the above-mentioned results and analysis, our DTLC is significantly superior to other methods on this complex data set.

V. CONCLUSION

In this article, we develop a novel DTLC approach for unsupervised visual domain adaptation. DTLC incorporates domain-invariant feature learning with category-discriminative information preserved and target pseudolabel refinement by label consistency into one framework. The discriminative transfer feature learning procedure would avoid the effects of feature distortion and facilitate transferring source knowledge to target effectively. The coarse-to-fine label refinement

procedure based on the between and within domain label consistency could further benefit classwise distributions alignment across the source and target. Thus, the positive feedback way between these two procedures significantly improves the final target classification accuracy. Comprehensive experiments on several visual cross-domain data sets show that DTLC significantly outperforms SOTA shallow and competitive deep domain adaptation methods.

REFERENCES

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

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.

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

[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, pp. 2261–2269.

[6] L. Zhang, W. Zuo, and D. Zhang, "LSDT: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1177–1191, Mar. 2016.

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

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

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

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

[11] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6103–6115, Dec. 2019.

[12] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2980–2989, Oct. 2011.

[13] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptationwith distribution matching machines," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2795–2802.

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

[15] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[16] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4058–4065.

[17] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 136–144.

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

[19] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3723–3732.

[20] M. Chen, K. Q. Weinberger, and J. C. Blitzer, "Co-training for domain adaptation," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Oct. 2011, pp. 2456–2464.

[21] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2007, pp. 601–608.

[22] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1682–1695, Jul. 2017.

[23] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Dec. 2013, pp. 769–776.

[24] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1410–1417.

[25] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2007, pp. 513–520.

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

[27] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[28] C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5552–5562, Dec. 2016.

[29] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2003, pp. 321–328.

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

[31] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, vol. 6, no. 7, pp. 2058–2065.

[32] 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, Jul. 2017.

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

[34] J. Liang, R. He, Z. Sun, and T. Tan, "Aggregating randomized clustering-promoting invariant projections for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1027–1042, May 2019.

[35] 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, Sep. 2017.

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

[37] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 2208–2217.

[38] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8004–8013.

[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: https://arxiv.org/abs/1412.3474

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, vol. 1, no. 2, pp. 2962–2971.

[41] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3934–3941.

[42] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[43] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 499–515.

[44] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," 2017, *arXiv:1702.05374*. [Online]. Available: https://arxiv.org/abs/1702.05374

[45] Q. Xiao, H. Luo, and C. Zhang, "Margin sample mining loss: A deep learning based method for person re-identification," 2017, *arXiv:1710.00478*. [Online]. Available: https://arxiv.org/abs/1710.00478

[46] B. Schölkopf *et al.*, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[47] A. Saha, P. Rai, H. Daumé, S. Venkatasubramanian, and S. L. DuVall, "Active supervised domain adaptation," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, Sep. 2011, pp. 97–112.
[48] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.
[49] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2007, pp. 137–144.
[50] 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, Dec. 2003.
[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[52] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, Sep. 2010, pp. 213–226.
[53] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn. (ICML)*. Jun. 2014, pp. 647–655.
[54] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694
[55] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5018–5027.
[56] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Trans. Comput.*, vol. C-24, no. 7, pp. 750–753, Jul. 1975.
[57] I. Jolliffe, *Principal Component Analysis*. Berlin, Germany: Springer, 2011, pp. 1094–1096.
[58] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2960–2967.
[59] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 443–450.
[60] J. Zhuo, S. Wang, W. Zhang, and Q. Huang, "Deep unsupervised convolutional domain adaptation," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 261–269.
[61] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Shuang Li received the Ph.D. degree in control science and engineering from the Department of Automation, Tsinghua University, Beijing, China, in 2018.

He was a Visiting Research Scholar with the Department of Computer Science, Cornell University, Ithaca, NY, USA, from November 2015 to June 2016. He is currently an Assistant Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. His main research interests include machine learning and deep learning, especially in transfer learning and domain adaptation.

Chi Harold Liu (SM'15) received the B.Eng. degree from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree from the Imperial College London, London, U.K., in 2010.

He is currently a Full Professor and the Vice Dean with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Before moving to Academia, he joined IBM Research–China, Beijing, as a Staff Researcher and a Project Manager, after working as a Post-Doctoral Researcher with the Deutsche Telekom Laboratories, Berlin, Germany, and a Visiting Scholar with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. He has published more than 90 prestigious conference and journal articles and holds more than 14 EU/U.S./U.K./China patents. His current research interests include the big data analytics, mobile computing, and deep learning.

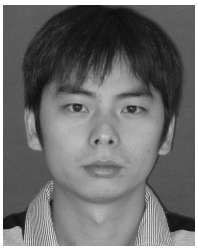
Dr. Liu is a fellow of IET.

Limin Su is currently pursuing the master's degree with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China.

Her current research interests include machine learning and transfer learning.

Binhui Xie is currently pursuing the master's degree with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China.

His research interests include computer vision and transfer learning.



Zhengming Ding (S'14–M'18) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA, in 2018.

He has been a Faculty Member with the Department of Computer, Information and Technology, Indiana University-Purdue University Indianapolis, Indianapolis, IN, USA, since 2018. His current research interests include transfer learning, multiview learning, and deep learning.

Dr. Ding received the National Institute of Justice Fellowship from 2016 to 2018. He was a recipient of the Best Paper Award by the SPIE 2016 and the Best Paper Candidate by the ACM MM 2017. He is currently an Associate Editor of *Journal of Electronic Imaging*.

C. L. Philip Chen (S'88–M'88–SM'94–F'07) graduated from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 1985.

He is currently a Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen is a fellow of the AAAS, IAPR, the Chinese Association of Automation (CAA), and HKIE, and a member of the Academia Europaea (AE), the European Academy of Sciences and Arts (EASA), and the International Academy of Systems and Cybernetics Science (IASCYS). He was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, and the IEEE Norbert Wiener Award for his contribution in systems and cybernetics, and machine learnings in 2018. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017. He was the President of the IEEE Systems, Man, and Cybernetics Society from 2012 to 2013. He is currently the Vice President of the CAA. He is also the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON CYBERNETICS.

Dapeng Wu (S'98–M'04–SM'06–F'13) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests include networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security.

Dr. Wu is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.