Discriminative Transfer Learning for Driving Pattern Recognition in Unlabeled Scenes

Liu Yang, *Member, IEEE*, Maoying Li, Chenyang Shen, Qinghua Hu, *Senior Member, IEEE*, Jia Wen, and Shujie Xu

***Abstract*—Driving pattern recognition based on features, such as GPS, gear, and speed information, is essential to develop intelligent transportation systems. However, it is usually expensive and labor intensive to collect a large amount of labeled driving data from real-world driving scenes. The lack of a labeled data problem in a driving scene substantially hinders the driving pattern recognition accuracy. To handle the scarcity of labeled data, we have developed a novel discriminative transfer learning method for driving pattern recognition to leverage knowledge from related scenes with labeled data to improve recognition performance in unlabeled scenes. Note that data from different scenes may have different distributions, which is a major bottleneck limiting the performance of transfer learning. To address this issue, the proposed method adopts a discriminative distribution matching scheme with the aid of pseudolabels in unlabeled scenes. It is able to reduce the intraclass distribution disagreement for the same driving pattern among labeled and unlabeled scenes while increasing the interclass distance among different patterns. Pseudolabels in unlabeled scenes are updated iteratively via an ensemble strategy that preserves the data structure while enhancing the model robustness. To evaluate the performance of the proposed method, we conducted comprehensive experiments on real-world parking lot datasets. The results show that the proposed method can substantially outperform state-of-the-art methods in driving pattern recognition.**

***Index Terms*—Driving pattern recognition, interclass separability, intraclass compactness, maximum mean discrepancy (MMD), transfer learning.**

Manuscript received October 18, 2019; revised February 8, 2020; accepted April 8, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61732011 and Grant 61702358, in part by the Beijing Natural Science Foundation under Grant Z180006, in part by the Key Scientific and Technological Support Project of Tianjin Key Research and Development Program under Grant 18YFZCGX00390, and in part by the Tianjin Science and Technology Plan Project under Grant 19ZXZNGX00050. This article was recommended by Associate Editor Y. S. Ong. *(Corresponding author: Qinghua Hu.)*

Liu Yang, Qinghua Hu, and Jia Wen are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Machine Learning, Tianjin University, Tianjin 300350, China (e-mail: yangliuyl@tju.edu.cn; huqinghua@tju.edu.cn; wenjia@tju.edu.cn).

Maoying Li is with the Research and Development Department, Automotive Data of China (Tianjin) Company Ltd., Tianjin 300393, China, and also with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: limaoying@catarc.ac.cn).

Chenyang Shen is with the Division of Medical Physics and Engineering, Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX 75390 USA (e-mail: chenyang.shen@utsouthwestern.edu).

Shujie Xu is with the China Automotive Technology and Research Center, China Automotive Technology and Research Center Company Ltd., Tianjin 300300, China (e-mail: xushujie@catarc.ac.cn).

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/TCYB.2020.2987632

I. Introduction

RECENTLY, driving pattern recognition, that is, identifying specific movements of cars, such as driving in a lane, turning left, parking, etc., has been extensively studied given its critical importance for self-driving vehicles and intelligent transportation systems [1]. The driving status, such as GPS, gear, and speed information, is important and can be used for driving pattern recognition, but it may be highly variable depending on driving scenes such as different parking lots in Fig. 1. Hence, massive labeled data are often required for accurate driving pattern recognition. However, it is almost impossible to collect sufficient labeled driving data from every driving scene in practice. Consequently, driving pattern recognition for unlabeled scenes becomes a challenging problem of central importance. It is generally assumed that each driving pattern, such as turning left should be highly correlated across related scenes, despite their large disagreements in distribution. Thus, driving information learned from labeled scenes can be helpful for general driving pattern recognition. Motivated by the success of transfer learning [2], [3], which infers labels in an unlabeled target domain by leveraging knowledge from auxiliary domains, we focus on effectively using information from labeled driving scenes to enhance driving pattern recognition for the myriad of the unlabeled scene.

As previously mentioned, an obvious obstacle for transfer learning efficacy is the distribution discrepancy among data collected from different scenes. Consider parking lot data[1] as an example, where a dataset records vehicle statuses from different parking lots at different instants. As shown in Fig. 1, there are many types of parking lots with varying structures, and driving data considerably vary due to different road layouts and other conditions. To illustrate this point, Fig. 2 was generated by the random selection of 10 000 samples from different parking lots, and shows the distributions of two important features, namely, vehicle speed and steering wheel angle, to perform pattern recognition on an identical scenario of driving along a straight line. The data distribution and feature values poorly agree, showing the diversity of driving features according to the scene, which is one of the major bottlenecks that hinders transfer learning.

Successful transfer learning highly depends on reducing discrepancy in data distributions for the same driving pattern appearing in different scenes while preserving the

[1]The parking lot data were collected by China Automotive Technology and Research Center Company Ltd.

Fig. 1. Different driving scenes (e.g., parking lots with varying structures).

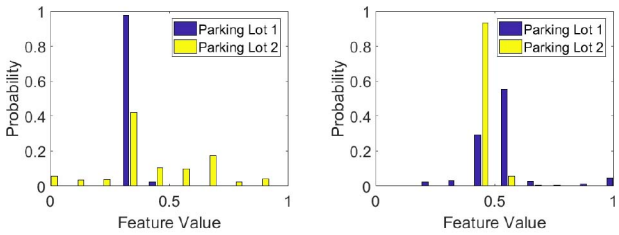


Fig. 2. Distributions of (a) vehicle speed and (b) steering wheel angle according to two parking lots are different.

original data structure. Several methods have been recently proposed to discover common feature representations while preserving important properties from original data over different domains [4]–[9]. However, these methods consider global distributions across all the samples, failing to account for the group structure belonging to different labels [10], [11]. In addition, it is not possible to evaluate intraclass disagreement in distributions for an unlabeled target domain unless pseudolabels are assigned, for which pseudolabels have been commonly employed. Pseudolabels are usually generated using conventional classifiers beforehand to guide data preprocessing for transfer learning [12]–[17], while the generated pseudolabels are not updated during transference. Moreover, these methods mainly consider the intraclass discrepancy, neglecting the dispersion across classes. In many challenging real problems, the performance of these classifiers is unsatisfactory and unreliable pseudolabels are thereby obtained, given the large discrepancy in distributions, undermining the effectiveness of transfer learning. A more suitable approach should gradually refine the pseudolabels by incorporating knowledge from labeled domains along with the transfer learning process. The refined labels may then provide more accurate group information in the target domain, enhancing the transfer learning efficacy.

To tackle all these issues and improve the transfer learning performance, we developed the proposed framework, discriminative transfer learning (DTL). The source and target data are projected onto a common space via discriminative transfer matching to enhance intraclass compactness, that is, affinity within the same class, and interclass separability, that is, dispersion among different classes. And then the majority voting method is used to iteratively refine the pseudolabels of target data. The proposed framework is illustrated in Fig. 3. The main contributions of this article can be summarized as follows.

1) DTL develops a novel transfer learning technique to address the driving pattern recognition problem for unlabeled scenes.
2) The proposed method incorporates the discriminative distribution matching which enhances intraclass compactness and interclass separability to reduce the discrepancy in distributions, that is, one of the bottleneck problems of transfer learning.
3) DTL iteratively refines the pseudolabels using an ensemble classifier to preserve the original data structure and enhance the DTL.

The remainder of this article is organized as follows. First, we briefly review the related work in Section II. The detailed design and implementation of DTL are provided in Sections III and IV, respectively. Section V presents the performance of DTL compared to state-of-the-art methods. Finally, we draw conclusions in Section VI.

II. Related Work

As we intend to apply transfer learning to driving pattern recognition, we briefly discuss the existing related methods.

Transfer learning [2] is inspired by the human ability to apply previous knowledge in different situations to solve new problems. It has achieved remarkable success in a wide range of applications, including text sentiment classification [18], [19]; image classification [20]–[23]; human activity classification [24], [25]; software defect classification [26]; and multilanguage text classification [27], [28]. Compared to traditional learning, transfer learning has a bottleneck regarding the discrepancy in data distributions between the source and target domains. Consider driving pattern recognition as an example. The driving status data are recorded in various scenes and under distinct conditions, usually leading to distinct data distributions. Thus, transfer learning should reduce the discrepancy among data distributions.

Remarkable research efforts have been devoted to unify data representation from different domains and thus improve the agreement among distributions while preserving important properties from the original data. For instance, the discriminative deep multimetric learning method [29] jointly learns multiple neural networks to characterize the correlation among different domains. Sharable and individual multiview metric learning [30] seeks for an individual distance metric for each view and a shared representation for different views. Note that these two methods were developed based on supervised information of similar and distinct pairs. Transfer component analysis (TCA) [4] learns the transfer components across domains using the maximum mean discrepancy (MMD) [31], [32], a common measure of the difference among domains. Transfer joint matching (TJM) [5] aims to enhance the agreement in data distributions by jointly matching features and reweighting the instances across domains. Geodesic flow kernel (GFK) [6]

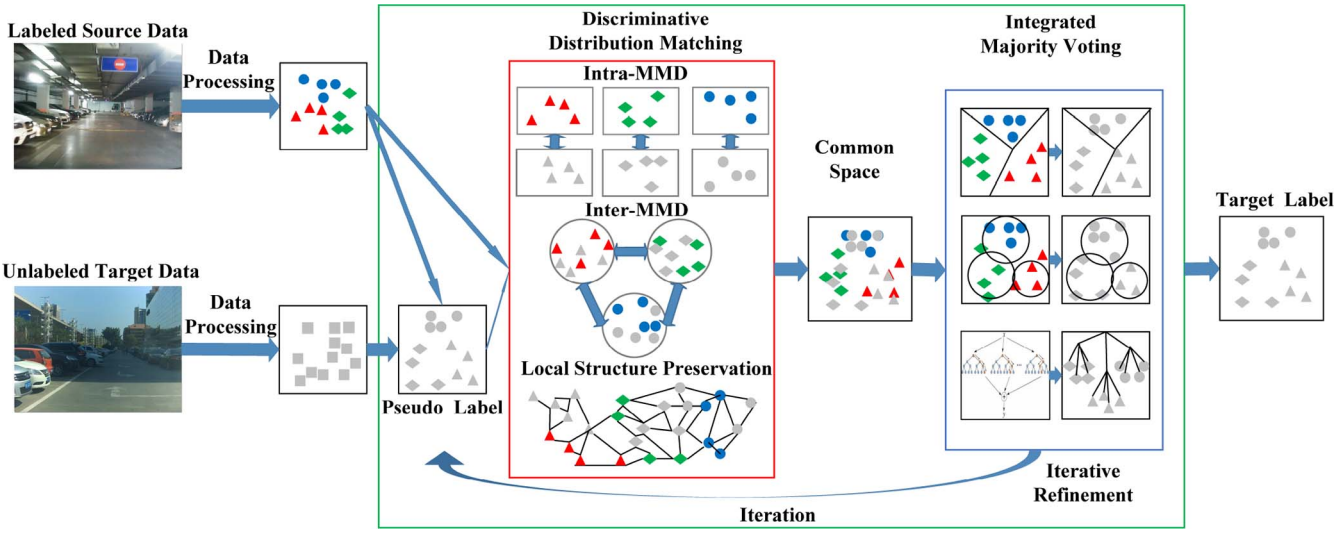


Fig. 3. Framework of the proposed DTL method. Source and target data can be projected onto a common space by the discriminative distribution matching, which can reduce the MMD within classes, increase the distance across classes, and preserve the local manifold structure. Then, the majority voting classification based on source data can be used to iteratively refine the labels of target data.

exploits low-dimensional structures and integrates different domains. Correlation alignment (CORAL) [7] minimizes the domain shift by aligning the second-order statistics from source to target distributions. Moreover, landmarks-based kernelized subspace alignment (LSA) [8] aims to select important landmarks in data and perform subspace alignment for interdomain discrepancy reduction. Domain adaptive neural networks (DANNs) [9] use simple neural-network models for domain adaptation in object recognition. They incorporate the MMD as regularization in the supervised learning process to reduce distribution disagreement between the source domains and target domain in the latent space. Although these transfer learning methods focus on enhancing the agreement in global data distributions between the source domains and target domain, they neglected the group structure and mixed data of different labels, leading to the global domain shift [10], [11].

According to [33], features from data samples with the same label across different domains should lie on a common subspace, called latent space or submanifold. Therefore, the agreement in intraclass distributions should be enforced for data from different domains to prevent the global domain shift. To this end, pseudolabels of samples in the target domain can be defined. Existing methods often rely on pseudolabels obtained by conventional classifiers. For instance, joint distribution adaptation (JDA) [12] and balanced distribution adaptation (BDA) [13] jointly adopt both the marginal and conditional distributions based on predicted pseudolabels. Adaptation regularization-based transfer learning (ARTL) [14] employs the MMD as a distance measure to perform marginal distribution adaptation. Manifold embedded distribution alignment (MEDA) [15] learns dynamic distribution alignment to quantitatively account for marginal and conditional distributions. Joint geometric and statistical alignment (JGSA) [16] projects source and target domain data onto two subspaces, where the geometric shift and distribution shift are simultaneously reduced. Stratified transfer learning (STL) [17] obtains pseudolabels for the target domain via majority voting. They generate pseudolabels before transfer learning, completely relying on conventional classifiers employed before data adaptation between different domains. The moving semantic transfer network (MSTN) [34] employs the AlexNet architecture while the centroid alignment is performed for deep features to reduce the discrepancy between the source domain and target domain.

However, these methods mainly consider the intraclass discrepancy, neglecting the dispersion across classes. The joint and discriminative domain adaptation (JDDA) method [35] takes advantage of the ResNet architecture, and focuses on the intraclass compactness and interclass separability of the source domains only. As both the source domains and target domain are expected for domain adaptation, the adequate data alignment could further improve learning efficacy. The consideration motivates the proposed DTL to alternatively and iteratively update pseudolabels and adapt inter- and intra-MMD on both the source and target domains to obtain better transfer learning performance.

III. Driving Pattern Recognition

In this section, we introduce the proposed DTL framework. First, we define the problem and describe the main idea of DTL. Then, the three major steps of DTL are detailed, namely, discriminative distribution matching, pseudolabel prediction, and iterative refinement.

A. Problem Statement

DTL transfers knowledge from labeled samples in a source domain to identify the patterns among samples in a target domain without label information. More specifically, domain $\mathcal{D}$ is composed of an m-dimensional feature space $\mathcal{X}$ and C-cardinality label set $\mathcal{Y}$, where $\mathbf{x} \in \mathcal{X}$ is a sample and

TABLE I
NOTATION

Symbol	Definition
$\mathcal{D}_s, \mathcal{D}_t$	source and target domains
n_s, n_t	number of source and target samples
$n = n_s + n_t$	total number of samples
$n^{(c)}$	number of samples from c-th class
$n^{(c,v)}$	number of samples from c-th and v-th classes
m, C	number of shared features and classes
k	number of subspace bases
β, α	regularization parameters
$X_s \in \mathbb{R}^{m\times n_s}$	source data matrix
$X_t \in \mathbb{R}^{m\times n_t}$	target data matrix
$A \in \mathbb{R}^{m\times k}$	transformation matrix
$H \in \mathbb{R}^{n\times n}$	centering matrix
$X^{(c)} \in \mathbb{R}^{m\times n^{(c)}}$	data matrix of c-th class
$M^{(c)} \in \mathbb{R}^{n^{(c)}\times n^{(c)}}$	intra-class MMD matrix of c-th class
$X^{(c,v)} \in \mathbb{R}^{m\times n^{(c,v)}}$	data matrix of c-th and v-th classes, $v \neq c$
$\Gamma^{(c,v)} \in \mathbb{R}^{n^{(c,v)}\times n^{(c,v)}}$	inter-class MMD matrix, $v \neq c$
L	graph Laplacian matrix

$y \in \mathcal{Y}$ is its label. For domain $\mathcal{D}$, task $\mathcal{T}$ is learning classifier $f(\cdot)$ based on $\mathcal{X}$ and $\mathcal{Y}$. Given a labeled source domain $\mathcal{D}_s = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n_s}, y_{n_s})\}$ and an unlabeled target domain $\mathcal{D}_t = \{\mathbf{x}_{n_s+1}, \ldots, \mathbf{x}_{n_s+n_t}\}$, we can form the source and target data matrices, $X_s = [\mathbf{x}_1, \ldots, \mathbf{x}_{n_s}] \in \mathbb{R}^{m\times n_s}$ and $X_t = [\mathbf{x}_{n_s+1}, \ldots, \mathbf{x}_{n_s+n_t}] \in \mathbb{R}^{m\times n_t}$, respectively. For $n = n_s + n_t$, $X = [X_s\ X_t] \in \mathbb{R}^{m\times n}$ is the data matrix combining all the samples in both the source and target domains. Assume that the feature and label spaces of the source and target domains are the same, that is, $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$, but their distributions differ. The goal is to learn classifier $f(\cdot)$ for the unlabeled data in target domain $\mathcal{D}_t$ based on information in source domain $\mathcal{D}_s$ by matching the distributions between the domains. Table I shows the notation adopted throughout this article.

B. Proposed DTL Framework

As shown in Fig. 3, the proposed DTL alternatively and iteratively updates the feature transformation and pseudolabels. Specifically, the intraclass discrepancy in distributions is minimized, whereas the interclass discrepancy is maximized, and the pseudolabels are refined throughout learning. The proposed iterative strategy provides adaptive refinement of data distributions and pseudolabels to eventually achieve accurate classification in the target domain. The major steps of DTL are detailed in the remainder of this section.

1) Discriminative Distribution Matching: We aim to reduce the discrepancy between domain distributions by feature transformation A, such that the joint expectations of the source and target domains suitably agree afterward. To reduce the discrepancy, we adopt the MMD [31], [32], which retrieves the distance between expectations of samples from the source and target domains in k-dimensional embedding as a distance measure between distributions. We further modify the MMD into two measures, namely, intra-MMD and inter-MMD, to further restrict the transformation, such that the intraclass and interclass distances can be minimized and maximized, respectively. In addition, a regularization term is included to ensure that the transformation preserves locality from the original domains. As the labels for the target domain are unknown, we use pseudolabels, as detailed in the following section.

Intra-MMD: The intraclass MMD is defined as

$$\begin{aligned}\ell_{\text{intra}} &= \sum_{c=1}^{C}\left\|\frac{1}{n_s^{(c)}}\sum_{\mathbf{x}_i\in\mathcal{D}_s^{(c)}}A^T\mathbf{x}_i - \frac{1}{n_t^{(c)}}\sum_{\mathbf{x}_j\in\mathcal{D}_t^{(c)}}A^T\mathbf{x}_j\right\|^2 \\ &= \sum_{c=1}^{C}\text{tr}(A^TX^{(c)}M^{(c)}(X^{(c)})^TA) \end{aligned} \tag{1}$$

where

$$(M^{(c)})_{ij} = \begin{cases} \frac{1}{n_s^{(c)}n_s^{(c)}}, \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{n_t^{(c)}n_t^{(c)}}, \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \frac{-1}{n_s^{(c)}n_t^{(c)}}, \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{x}_j \in \mathcal{D}_s^{(c)}, \mathbf{x}_i \in \mathcal{D}_t^{(c)} \end{cases} \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

and tr$(\cdot)$ denotes the trace. $\mathcal{D}_s^{(c)} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathcal{D}_s \wedge y_i = c\}$ is the set of samples belonging to class c in the source data, y_i is the true label of $\mathbf{x}_i$ in the source domain, and $n_s^{(c)} = |\mathcal{D}_s^{(c)}|$. Correspondingly, $\mathcal{D}_t^{(c)} = \{\mathbf{x}_j : \mathbf{x}_j \in \mathcal{D}_t \wedge \hat{y}_j = c\}$ is the set of samples belonging to class c in the target data, $\hat{y}_j$ is the predicted pseudolabel of $\mathbf{x}_j$ in the target domain, and $n_t^{(c)} = |\mathcal{D}_t^{(c)}|$. $X^{(c)} \in \mathcal{D}_s^{(c)} \cup \mathcal{D}_t^{(c)}$. For $n^{(c)} = n_s^{(c)} + n_t^{(c)}$, $X^{(c)} \in \mathbb{R}^{m\times n^{(c)}}$.

$A \in \mathbb{R}^{m\times k}$ is the transformation matrix and MMD matrix $M^{(c)} \in \mathbb{R}^{n^{(c)}\times n^{(c)}}$ includes the class information. By minimizing (1), the intraclass distributions across domains are clustered under the new representation, $A^TX^{(c)}$.

Inter-MMD: The interclass MMD is defined as

$$\begin{aligned}\ell_{\text{inter}} &= \sum_{c=1}^{C}\sum_{v=1,v\neq c}^{C}\left\|\frac{1}{n^{(c)}}\sum_{\mathbf{x}_i\in\mathcal{D}^{(c)}}A^T\mathbf{x}_i - \frac{1}{n^{(v)}}\sum_{\mathbf{x}_j\in\mathcal{D}^{(v)}}A^T\mathbf{x}_j\right\|^2 \\ &= \sum_{c=1}^{C}\sum_{v=1,v\neq c}^{C}\text{tr}\left(A^TX^{(c,v)}\Gamma^{(c,v)}\left(X^{(c,v)}\right)^TA\right) \end{aligned} \tag{3}$$

where

$$(\Gamma^{(c,v)})_{ij} = \begin{cases} \frac{1}{n^{(c)}n^{(c)}}, \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}^{(c)} \\ \frac{1}{n^{(v)}n^{(v)}}, \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}^{(v)} \\ \frac{-1}{n^{(c)}n^{(v)}}, \begin{cases} \mathbf{x}_i \in \mathcal{D}^{(c)}, \mathbf{x}_j \in \mathcal{D}^{(v)} \\ \mathbf{x}_j \in \mathcal{D}^{(c)}, \mathbf{x}_i \in \mathcal{D}^{(v)} \end{cases} \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

$\mathcal{D}^{(c)} = \{\mathbf{x}_i : \mathbf{x}_i \in \{\mathcal{D}_s \cup \mathcal{D}_t\} \wedge \{y_i = c \cup \hat{y}_i = c\}\}$ is the set of samples belonging to class c in the source and target domains, y_i is the true label of $\mathbf{x}_i$, $\hat{y}_i$ is the pseudolabel of $\mathbf{x}_i$, and $n^{(c)} = |\mathcal{D}^{(c)}|$. Correspondingly, $\mathcal{D}^{(v)} = \{\mathbf{x}_j : \mathbf{x}_j \in \{\mathcal{D}_s \cup \mathcal{D}_t\} \wedge \{y_j = v \cup \hat{y}_j = v\}\}$ is the set of samples belonging to class v, $n^{(v)} = |\mathcal{D}^{(v)}|$, and $X^{(c,v)} \in \mathcal{D}^{(c)} \cup \mathcal{D}^{(v)}$. For $n^{(c,v)} = n^{(c)} + n^{(v)}$, $X^{(c,v)} \in \mathbb{R}^{m\times n^{(c,v)}}$. Matrix $\Gamma^{(c,v)} \in \mathbb{R}^{n^{(c,v)}\times n^{(c,v)}}$ includes the interclass information. Note that by maximizing (3) such that $-\ell_{\text{inter}}$ is minimized, the distributions among different classes are maximally separated.

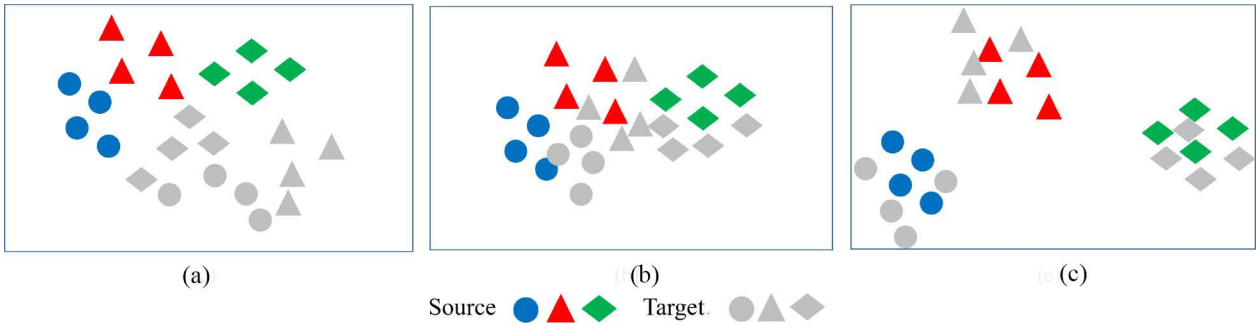


Fig. 4. (a) Original dataset and data transformation with (b) traditional distribution matching and (c) proposed discriminative matching.

Regularization Term: Locality preserving regularization ensures that two data points, $\mathbf{x}_i$ and $\mathbf{x}_j$, that are close to each other in the feature space remain close after transformation. This is also known as the local invariance assumption [36] and has been studied intensively in manifold learning. In addition, it has been successfully applied in diverse fields, such as semisupervised learning [37] and matrix factorization [38].

Like other studies [15], we model the local structure of the feature space by constructing a ε-nearest neighbor (NN) graph representing the n instances as vertices. The graph is constructed by assigning edges between each instance and its ε-nearest instances. Edge weights are computed by adopting the heat kernel with self-tuning for parameter σ [39]. Let W represent the constructed graph. Each weight $w_{i,j}$ is computed as $= w_{j,i} = \exp([-\|\mathbf{x}_i - \mathbf{x}_j\|^2]/\sigma)$ only if an edge is assigned between instances $\mathbf{x}_j$ and $\mathbf{x}_i$. Otherwise, $w_{i,j} = 0$. Then, regularization can be formulated as

$$\sum_{i,j=1}^{n} w_{i,j} \left\| A^T\mathbf{x}_i - A^T\mathbf{x}_j \right\|_2^2 = \mathrm{tr}\left(A^T X L X^T A\right). \tag{5}$$

$L = D - W$ is the graph Laplacian of W, where $D = [d_{i,i}]$ is a diagonal matrix with each diagonal element being computed by the corresponding column sum of W. To formulate both W and L, it is not necessary to know the label information of the instances, and therefore all instances $X = [X_s, X_t]$ in the feature space can be included. Minimizing (5) enforces optimality of A to preserve the local geometric structure, that is, mappings $A^T\mathbf{x}_i$ and $A^T\mathbf{x}_j$ should be close to each other if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar. Furthermore, (5) becomes a convex optimization problem if L is a positive-semidefinite matrix. During the calculation of W, we set $w_{i,j} = w_{j,i}$ for W to be symmetrical, guaranteeing that graph Laplacian matrix L is positive semidefinite. Regularization term ℓ_{reg} is given by

$$\ell_{\mathrm{reg}} = \mathrm{tr}\left(A^T X L X^T A\right). \tag{6}$$

Now, we can incorporate data fidelity and smoothness regularization terms to formulate the proposed transfer learning model. Like in [4], orthogonal constraint $A^T X H X^T A = I$ is introduced to eliminate the redundancy in transformation, where $H = I - (1/n)\mathbf{1}$ is the centering matrix

$$\begin{aligned} &\min_{A} \ \ell_{\mathrm{intra}} - \alpha \ell_{\mathrm{inter}} + \beta \ell_{\mathrm{reg}} \\ &\text{s.t.} \ \ A^T X H X^T A = I \end{aligned} \tag{7}$$

where α and β are parameters that balance the contributions of their corresponding terms. This model can learn the optimal mapping function A that can identify and utilize the correlations among labels while preserving the local geometric structure among instances in the feature space. Therefore, the proposed algorithm is expected to be useful for label prediction during transfer learning.

A comparison with the existing matching methods is shown in Fig. 4. Unlike the traditional distribution alignment methods that only match the source and target distributions within classes, our method can increase the distance across different classes.

2) Pseudolabel Prediction: At each iteration, classification is performed on data after feature transformation A is fixed. In addition, we update the pseudolabels for the target domain by integrating the classification results from various classifiers trained on the source domain via majority voting. It is expected that majority voting improves reliability to achieve better results compared to the use of individual classifiers, as numerically justified in Section V.

Let $\hat{y}_j (j = n_s + 1, \ldots, n_s + n_t)$ denote the result of majority voting classification on $\mathbf{x}_j$ and $f_r(j)$ denote the pseudolabel prediction of the jth sample by the rth classifier, $f_r(\cdot)$. Then

$$\hat{y}_j = \begin{cases} \mathrm{majority}(f_r(j), r), & \text{if majority holds} \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

where $r \in \{1, 2, \ldots, R\}$ denotes the index of the classifier. We can use any type of classifier in the proposed framework. In this article, we adopted the widely used 1-NN [40], support vector machine (SVM) [41], and random forest (RF) [42] as classifiers given their effectiveness and ease of implementation.

3) Iterative Refinement: With the updated pseudolabels, DTL determines a new feature transformation to reduce the intra-MMD and increase the inter-MMD. The corresponding dataflow is illustrated in Fig. 3, where the feature transformation and pseudolabels are alternatively updated to refine the results until convergence.

IV. Transfer Learning Algorithm

A. Proposed DTL Algorithm

At each iteration, the data transformation to minimize the discrepancy in data distributions should be determined. It is formulated as the optimization problem in (7). In this section,

Algorithm 1: DTL for Driving Pattern Recognition

Input : Source domain $\mathcal{D}_s = \{(\mathbf{x}_i, y_i)_{i=1}^{n_s}\}$, target domain $\mathcal{D}_t = \{\{\mathbf{x}_j\}_{j=n_s+1}^{n_s+n_t}\}$, parameters α and β.
Output: Predicted labels of target data.

1 Train several classifiers on $\mathcal{D}_s$, and obtain majority voting result to initialize pseudo-labels $\{\hat{y}_j\}_{j=n_s+1}^{n_s+n_t}$ of target data in $\mathcal{D}_t$ by Eq. (8).
2 **repeat**
3 Construct intra-MMD matrices $M^{(c)}$ using Eq. (2).
4 Construct inter-MMD matrices $\Gamma^{(c,v)}$ using Eq. (4).
5 Solve generalized eigendecomposition in Eq. (12) and select the k smallest eigenvectors to construct adaptation matrix A.
6 Train several classifiers on $\{(A^T\mathbf{x}_i, y_i)\}_{i=1}^{n_s}$ and obtain the majority voting result to update pseudo-labels $\{\hat{y}_j\}_{j=n_s+1}^{n_s+n_t}$ of target data $\{A^T\mathbf{x}_j\}_{j=n_s+1}^{n_s+n_t}$ using Eq. (8).
7 **until** *convergence*;
8 Return target labels.

we derive the numerical algorithm to solve this optimization problem.

The objective function in (7) can be written as

$$\sum_{c=1}^{C} \mathrm{tr}\left(A^T X^{(c)} M^{(c)} \left(X^{(c)}\right)^T A\right) + \beta \mathrm{tr}\left(A^T X L X^T A\right) - \alpha \sum_{c=1}^{C} \sum_{v=1, v\neq c}^{C} \mathrm{tr}\left(A^T X^{(c,v)} \Gamma^{(c,v)} (X^{(c,v)})^T A\right). \quad (9)$$

Equation (9) can be rewritten as $\mathrm{tr}(A^T \Theta A)$, where Θ is given by

$$\Theta = \sum_{c=1}^{C} X^{(c)} M^{(c)} (X^{(c)})^T + \beta X L X^T - \alpha \sum_{c=1}^{C} \sum_{v=1, v\neq c}^{C} X^{(c,v)} \Gamma^{(c,v)} (X^{(c,v)})^T. \quad (10)$$

According to the constrained optimization, $\Phi = \mathrm{diag}(\phi_1, \ldots, \phi_k) \in \mathbb{R}^{k\times k}$ is the Lagrange multiplier, and we derive the Lagrange function for (7) as

$$\mathcal{O} = \mathrm{tr}\left(A^T \Theta A\right) + \sum_{c=1}^{C} \mathrm{tr}\left(\left(I - A^T X H X^T A\right)\Phi\right). \quad (11)$$

Setting $(\partial \mathcal{O}/\partial A) = 0$, we obtain generalized eigendecomposition

$$\Theta A = \left(\sum_{c=1}^{C} X H X^T\right) A \Phi. \quad (12)$$

Finally, the problem of finding optimal adaptation matrix A is reduced to solving (12) for the k smallest eigenvectors. The procedure adopted by DTL is described in Algorithm 1.

B. Complexity Analysis

Most of the computational burden of DTL comes from three operations: 1) MMD matrix construction (lines 3 and 4 in Algorithm 1); 2) eigendecomposition (line 5 in Algorithm 1); and 3) the ensemble classifier (line 6 in Algorithm 1). To construct the intra-MMD and inter-MMD matrices, the complexity is $\mathcal{O}(C(n^{(c)})^2 + C^2(n^{(c,v)})^2)$. As $n^{(c)} < n^{(c,v)}$, it can be reduced to $\mathcal{O}(C^2(n^{(c,v)})^2)$. For solving eigendecomposition, we should calculate matrix Θ, and the complexity is $\mathcal{O}(mC(n^{(c)})^2 + mC^2(n^{(c,v)})^2 + mn^2)$, which can be reduced to $\mathcal{O}(mC^2(n^{(c,v)})^2 + mn^2)$. The complexity of eigendecomposition per iteration is $\mathcal{O}(m^3)$. In our experiments, 1-NN, SVM, and RF were used as pseudolabel classifiers. Their complexities are $\mathcal{O}(mn)$, $\mathcal{O}(n^3)$, and $\mathcal{O}(mnt)$, respectively, where t is the depth of the tree. Therefore, the overall computational complexity of Algorithm 1 is $\mathcal{O}(\tau(mC^2(n^{(c,v)})^2 + mn^2 + n^3 + m^3 + mnt))$, where τ is the number of iterations.

V. Experiments and Evaluation

We conducted extensive experiments on driving pattern recognition to evaluate the proposed DTL approach.

A. Data Preparation

We applied the proposed DTL to identify driving patterns in different parking lot scenes. The detailed data collection and preprocessing steps are summarized in the following.

1) Data Collection: The driving data were collected from different parking lots with distinct structures and road conditions. Each vehicle for data acquisition was equipped with several sensors, as shown in Fig. 5. The data attributes include Frame ID, Turn Light, Vehicle Speed, Steering Wheel, Brake, Brake Information, Engine Speed, Accelerated Speed, Gear, Mileage, Oil Consumption, Date Time, Longitude, Latitude, Altitude, Angle, and GPS Speed. We aimed to recognize the 16 different driving patterns listed in Table II. Five datasets were collected from five parking lots *A–E* collecting 10 047, 9157, 8924, 9765, and 8436 samples, respectively. As shown in Fig. 1, different parking lots have very different data distributions, increasing the divergence between the source and target domains.

2) Data Preprocessing: Based on the original data, we performed preprocessing before adapting the proposed DTL framework for driving pattern recognition. From 36 attributes in the original data, we selected 13 while removing those highly correlated among all instances in one or more domains, as they do not add discriminative information. Then, the selected attributes were encoded for subsequent processing. For instance, character attributes, such as *Gear* with possible states *D, N, N/A*, and *R*, were represented using one-hot encoding. Featurewise normalization was then performed for each dataset individually. The datasets from different parking lots have distinct distributions, as shown in Fig. 2. Hence, each dataset from one parking lot was considered as belonging to one scene (domain).

To investigate more detailed information during knowledge transfer for driving scenes, we applied DTL to two scenarios, either one or multiple source domains. We use notation

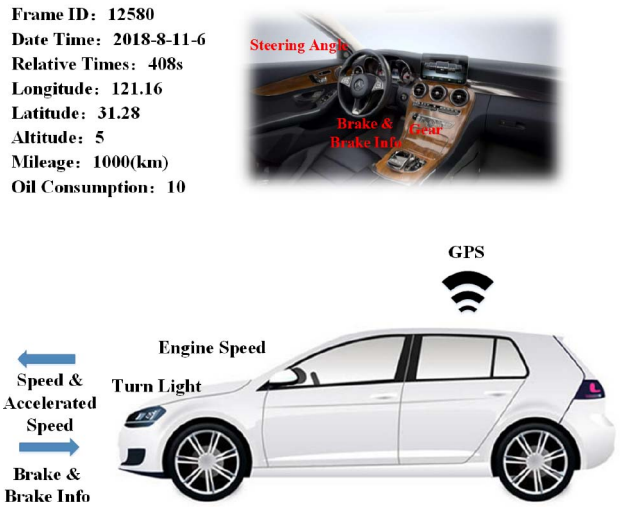


Fig. 5. Driving status features.

TABLE II
DRIVING PATTERNS IN PARKING LOT DATASETS

Class	Pattern
1	Driving in lane outside parking lot
2	Going straight at intersection outside parking lot
3	Turning left at intersection outside parking lot
4	Turning right at intersection outside parking lot
5	Entering parking lot
6	Driving in lane looking for parking spot
7	Going straight at intersection looking for parking spot
8	Turning left at intersection looking for parking spot
9	Turning right at intersection looking for parking spot
10	Parking
11	Leaving parking spot
12	Driving in lane looking for exit
13	Going straight at intersection looking for exit
14	Turning left at intersection looking for exit
15	Turning right at intersection looking for exit
16	Leaving parking lot

$A \rightarrow B$ to indicate labeling of the driving scene in domain B using labeled domain A. We conducted 19 learning tasks, with 16 having one source domain and three having four source domains.

B. Comparison Methods

We compared the proposed DTL approach to 17 classifiers.

1) 1-NN [40] assigns each testing sample to the class most common via its NN.
2) SVM [41] divides the samples of the separate categories by a clear margin that is as wide as possible.
3) RF [42] is an ensemble learning method which constructs a multitude of decision trees and outputs the class with the individual trees.
4) TCA [4] learns the transfer components across domains using MMD, then data distributions in different domains are close to each other.
5) GFK [6] exploits low-dimensional structures and changes the geometric and statistical properties from the source to the target domain.
6) JDA [12] aims to jointly adapt both the marginal distribution and conditional distribution in a dimensionality reduction procedure.
7) TJM [5] aims to enhance the agreement in data distributions by jointly matching features and reweighting the instances across domains.
8) ARTL [14] employs the MMD as distance measure to perform marginal distribution adaptation.
9) CORAL [7] minimizes the domain shift by aligning the second-order statistics from source and target distributions.
10) LSA [8] aims to select important landmarks and perform subspace alignment for reducing interdomain discrepancy.
11) JGSA [16] projects all data to two subspaces, where the geometric shift and distribution shift are simultaneously reduced.
12) BDA [13] can adaptively leverage the importance of the marginal and conditional distribution discrepancies.
13) STL [17] obtains pseudolabels for the target domain via majority voting and then performs intraclass knowledge transfer.
14) MEDA [15] learns dynamic distribution alignment to quantitatively account for marginal and conditional distributions.
15) DANNs [9] uses simple neural-network models for domain adaptation in object recognition.
16) MSTN [34] employs the AlexNet architecture while the centroid alignment is performed for deep features.
17) The instance-based JDDA method [35] takes advantage of the ResNet architecture.

The 1-NN, SVM, and RF classifiers can be considered as conventional methods, whereas the others are transfer learning approaches. They aim to reduce the difference between the source and target domains. Then, they train the classifier on the labeled source data, and test it on the unlabeled target data. The codes for the comparison methods are available online.

C. Evaluation

Under our experimental setup, the optimal parameters cannot be obtained using cross-validation, as labeled and unlabeled data are sampled from different distributions. Thus, we evaluated all methods by heuristically searching the parameter space for the optimal settings and report the best results for each method. We used the widely used classification accuracy on test data as evaluation measure [6], [4]

$$\text{Accuracy} = \frac{|\mathbf{x}_j : \mathbf{x}_j \in \mathcal{D}_t \wedge \hat{y}_j = y_j|}{|\mathbf{x}_j : \mathbf{x}_j \in \mathcal{D}_t|} \tag{13}$$

where $\mathcal{D}_t$ is the set of test data, y_j is the truth label of $\mathbf{x}_j$, and $\hat{y}_j$ is the label predicted by the classification algorithm. We executed each algorithm ten times with different random initializations and obtained the average results.

D. Results and Analysis

1) Effectiveness of Intraclass and Interclass Transference: To verify the effectiveness of DTL from the distribution

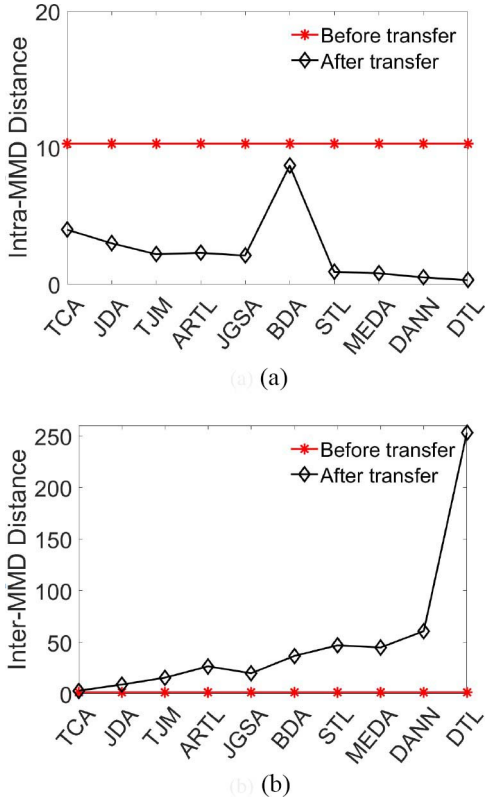


Fig. 6. (a) Intra-MMD and (b) inter-MMD of all samples on the dataset $ACDE \rightarrow B$.

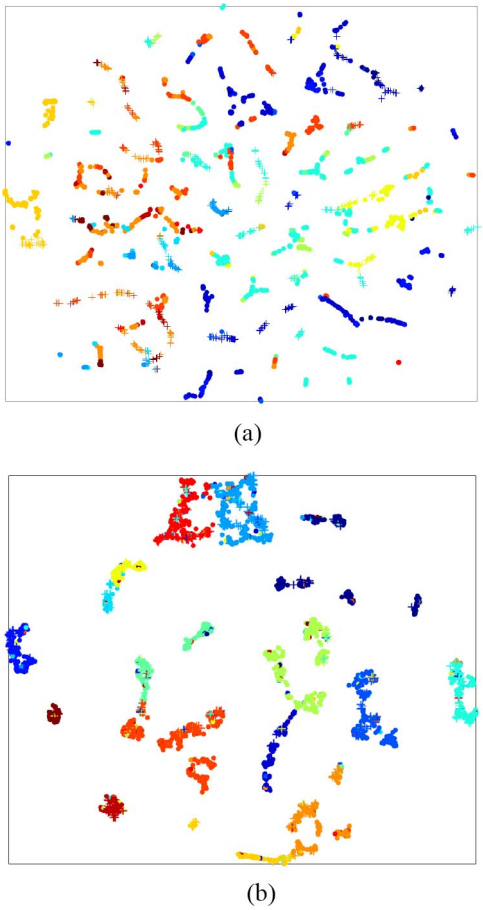


Fig. 7. (a) Data visualization before transformation (intra-MMD, 10.3; inter-MMD, 1.6). (b) Data visualization after transformation (intra-MMD, 0.3; inter-MMD, 253.3) on the dataset $ACDE \rightarrow B$. "$\circ$" and "$+$" represent the source and target domains, respectively, different colors represent different categories.

distance, we compared it to nine methods that can reduce the MMD between source and target domains. Take the $ACDE \rightarrow B$ dataset as an example. The intra-MMD and inter-MMD of all methods are shown in Fig. 6. In addition, the t-distributed stochastic neighbor embedding of all data points before and after transformation is depicted in Fig. 7(a) and (b), respectively. "$\circ$" and "$+$" represent data from $ACDE$ and B, and different colors represent different classes. Points in the same class from different domains become closer after transformation. Compared with other methods, DTL can substantially reduce the discrepancy between the source and target domains, and increase the interclass distance, because DTL reduces the difference within each category and iteratively refines the pseudolabels using the ensemble strategy.

To illustrate the interclass discrepancy reduction using DTL, the intra-MMD and accuracy are listed in Table III. Most intra-MMD values are reduced by DTL, and the classification performance improves. In addition, the inter-MMD for TCA, JDA, STL, and DTL is shown in Fig. 8. Compared with other methods, most of the values of DTL increase. By iteratively updating the pseudolabels using the ensemble method, DTL can reduce the intraclass divergence and increase the interclass distance at every iteration to improve the classification performance.

To verify the effectiveness of DTL by embedding similarity, we computed the 10-NN similarity matrix on embedding $A^T X$ obtained from DTL. To better demonstrate the results, we only selected four classes and 25 samples per domain. The first 100 samples and the last 100 samples comprised the source and target domains, respectively. We also constructed the similarity matrices for TCA, JDA, and STL using their optimal parameter settings. The similarity matrices are depicted in Fig. 9, where the diagonal and anti-diagonal blocks indicate intraclass similarity within and across domains, respectively, whereas the other blocks indicate interclass similarity. DTL achieves higher intraclass and lower interclass similarity both within and across domains.

2) Parameter Sensitivity: The DTL approach involves three model parameters: 1) number k of subspace bases; 2) Laplacian regularization parameter β; and 3) interclass regularization parameter α. We conducted a sensitivity analysis to validate the DTL optimal performance under several parameter values. We randomly selected $A \rightarrow D$, $D \rightarrow B$, and $ACDE \rightarrow B$ to illustrate the parameter sensitivity analysis. We observed similar trends on all other datasets but do not report their results here due to space constraints.

Hyperparameter tuning is commonly needed in many other state-of-the-art methods. For instance, JDA [12] and STL [17] used as benchmarks in this article have parameters (♯subspace bases and regularization parameters) that need to be adjusted in order to achieve optimal performance. Similar to the compared methods, we ran DTL while varying k, β, and α, and

TABLE III
INTRA-MMD AND ACCURACY (%, IN PARENTHESES) FROM ORIGINAL AND TRANSFORMED SPACES ON THE DATASET *ACDE* → *B* WITH 16 CLASSES

Class	Original	TCA	JDA	TJM	ARTL	JGSA	BDA	STL	MEDA	DANN	DTL
1	1.66 (100)	0.00 (100)	0.01 (76)	0.15 (71)	0.00 (71)	0.00(76)	0.02 (76)	0.74 (100)	0.00 (29)	0.01 (73)	0.04 (88)
2	9.66 (55)	0.05 (90)	0.64 (100)	2.84 (100)	0.74 (99)	2.05 (91)	1.70 (100)	0.20 (100)	0.02 (98)	0.02(100)	0.10 (100)
3	1.52 (17)	0.02 (61)	0.59 (12)	0.30 (46)	0.36 (0)	0.06 (34)	0.07 (32)	1.13 (5)	0.50 (0)	0.40(0)	0.14 (12)
4	0.04 (86)	0.00 (43)	0.01 (100)	0.14 (79)	0.02 (0)	0.03 (93)	0.01 (36)	1.47 (71)	0.13 (0)	0.13 (0)	0.04 (71)
5	4.46 (65)	0.01 (56)	1.70 (74)	0.71 (76)	0.00 (68)	0.04 (57)	4.50 (66)	1.50 (71)	0.00 (60)	0.00 (68)	0.03 (74)
6	0.57 (30)	0.00 (65)	0.05 (78)	0.21 (83)	0.00 (43)	0.11 (48)	0.12 (52)	0.93 (65)	0.00 (65)	0.00(65)	0.03 (65)
7	4.33 (97)	0.34 (100)	0.06 (96)	8.23 (95)	1.43 (100)	0.13 (97)	0.14 (97)	0.02 (95)	0.17 (99)	0.16 (96)	0.05 (100)
8	0.13 (67)	0.00 (100)	0.01 (89)	0.06 (89)	0.01 (78)	0.01 (89)	0.03 (89)	0.81 (89)	0.00 (44)	0.00 (79)	0.08 (89)
9	0.18 (55)	0.01 (0)	0.01 (77)	0.98 (80)	0.33 (0)	0.01 (100)	0.00 (84)	1.44 (80)	1.02 (0)	0.94 (0)	0.02 (89)
10	0.06 (68)	0.00 (83)	0.02 (89)	1.67 (91)	0.01 (40)	0.04 (100)	0.12 (83)	1.04 (79)	0.00 (58)	0.00(65)	0.04 (81)
11	3.52 (93)	0.03 (89)	1.31 (86)	1.14 (93)	0.00 (82)	0.63 (91)	0.67 (96)	1.71 (88)	0.00 (67)	0.00 (74)	0.14 (93)
12	1.98 (66)	1.44 (92)	1.05 (83)	2.21 (67)	1.69 (75)	2.17 (72)	0.17 (72)	0.14 (71)	0.10 (84)	0.11 (92)	0.20 (94)
13	8.93 (44)	0.06 (50)	0.01 (79)	3.89 (79)	0.55 (56)	0.45 (68)	0.04 (86)	0.05 (80)	0.01 (64)	0.01(62)	0.04 (77)
14	0.07 (19)	0.00 (25)	0.00 (13)	1.02 (19)	0.01 (0)	0.09 (56)	0.01 (31)	0.18 (13)	0.04 (0)	0.04(0)	0.02 (44)
15	1.02 (55)	0.00 (65)	0.02 (75)	1.72 (95)	0.03 (0)	0.04 (95)	0.02 (95)	0.91 (75)	0.16 (0)	0.17 (0)	0.11 (70)
16	0.04 (16)	0.00 (20)	0.01 (48)	0.16 (48)	0.00 (28)	0.01 (44)	0.01 (44)	1.46 (20)	0.00 (20)	0.00 (20)	0.06 (20)
Average	2.39 (58)	0.12 (65)	0.34 (73)	1.59 (75)	0.33 (46)	0.37 (74)	0.48 (71)	0.86 (69)	0.13 (43)	0.12 (50)	0.07 (76)

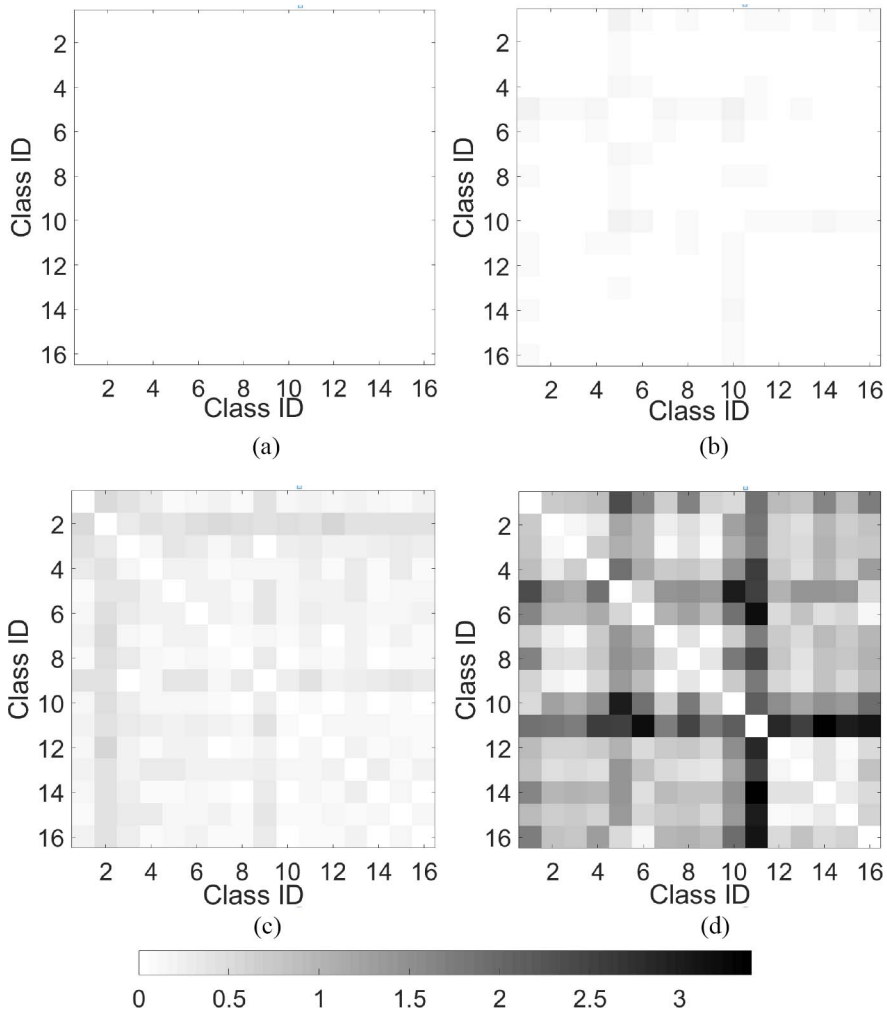


Fig. 8. Inter-MMD for 16 classes obtained from (a) TCA, (b) JDA, (c) STL, and (d) DTL on the dataset *ACDE* → *B*.

obtained the classification accuracy according to the different values, as shown in Fig. 10. To demonstrate that our model can achieve optimal performance under varying parameter values, the best results of other comparison methods are depicted as dashed lines. From Fig. 10(a), we chose $k \in [8, 13]$ for our experiments. Fig. 10(b) shows the classification accuracy according to the Laplacian parameter β, where the Laplacian constraint results useful to retain the local manifold structure in DTL, and our model generally outperforms the other methods in a wide range $\beta \in [10^{-6}, 10^{-1}]$. Fig. 10(c) shows the classification accuracy according to interclass regularization parameter α, where the inter-MMD should be increased. Again, our model outperforms the comparison methods in a wide range $\alpha \in [10^{-7}, 10^{-2}]$.

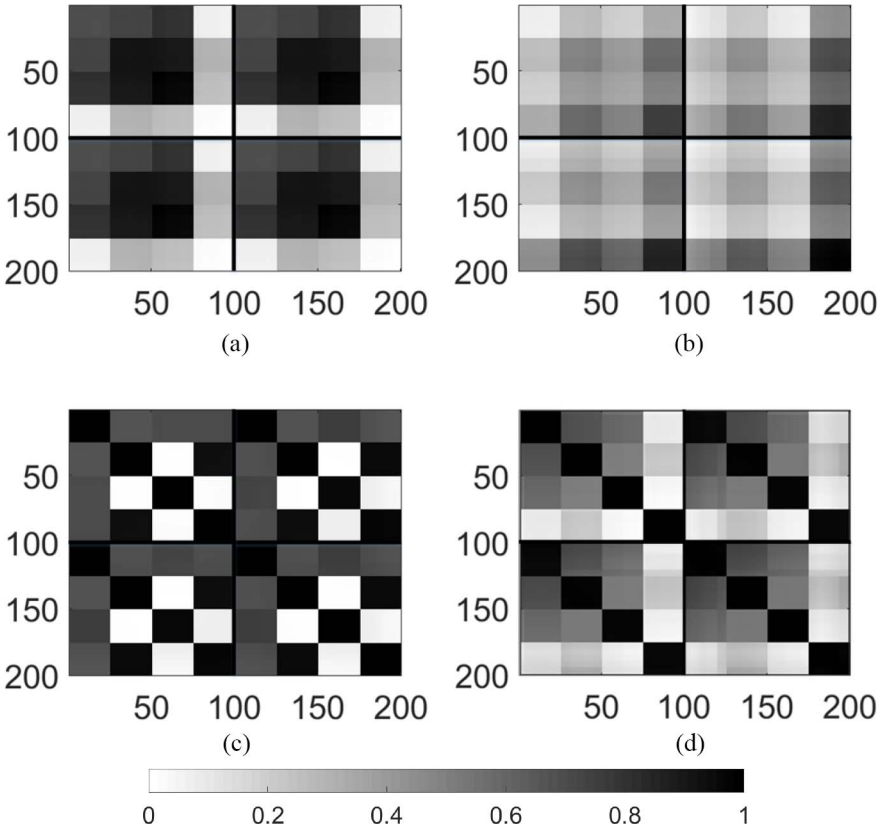


Fig. 9. Similarity matrices obtained from (a) TCA, (b) JDA, (c) STL, and (d) DTL embeddings on the dataset $ACDE \rightarrow B$.

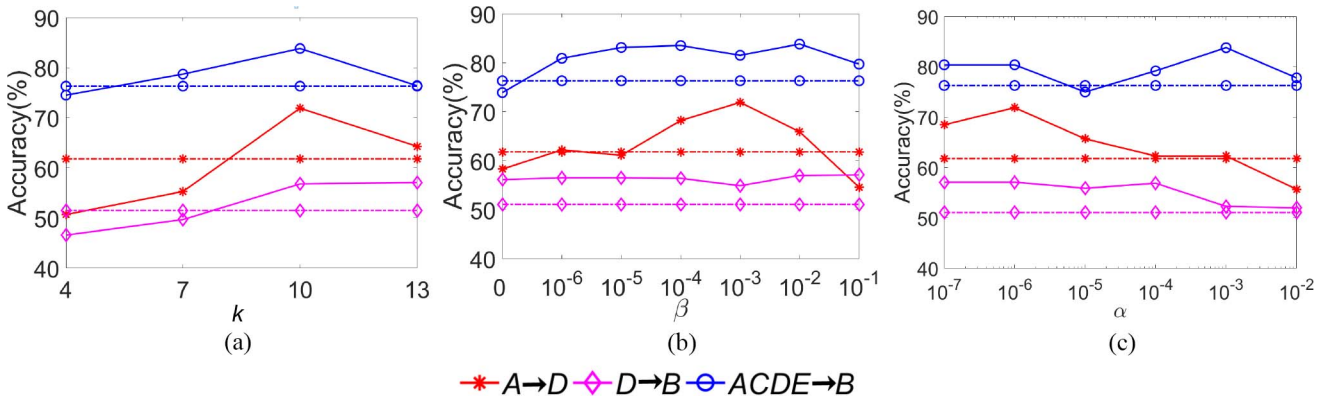


Fig. 10. Parameter sensitivity of DTL on three datasets (dashed lines indicate the best results from comparison methods). (a) ♯subspace bases k. (b) Laplacian regularization parameter β. (c) Interclass regularization parameter α.

We expect it is possible to perform hyperparameter tuning for the proposed method given a real-word dataset. Similar to the setup in our experiments, we should be able to tune the hyperparameters based on the performance in the source domain. A similar strategy has also been applied to adjust hyperparameters in [34] and [43]. For example, we compare the parameter sensitivity of k from A to other domains $A \rightarrow B$, $A \rightarrow C$, and $A \rightarrow D$. As shown in Fig. 11, we observe the trends of performance in the source and target domains with respect to different hyperparameter selections are consistent with each other, we are comfortable to use validation performance in the source domain as an indicator of performance in the target domain in the hyperparameter tuning process.

3) Algorithm Convergence: To investigate the convergence of the proposed DTL algorithm, we calculated the MMD according to the number of iterations, as shown in Fig. 12. The results were obtained from the $A \rightarrow D$, $D \rightarrow B$, and $ACDE \rightarrow B$ datasets. As the number of iterations increases, the MMD decreases, and accuracy increases. Hence, DTL exhibits good stability after a few iterations, suggesting its fast convergence.

4) Time Complexity: We verified the time complexity by running the evaluated transfer learning methods on the parking lot datasets $A \rightarrow D$, $D \rightarrow B$, and $ACDE \rightarrow B$. The running time and accuracy results are listed in Table IV. Among all comparison methods, the running times of TCA, GFK, CORAL, LSA, and STL are relatively short, but their

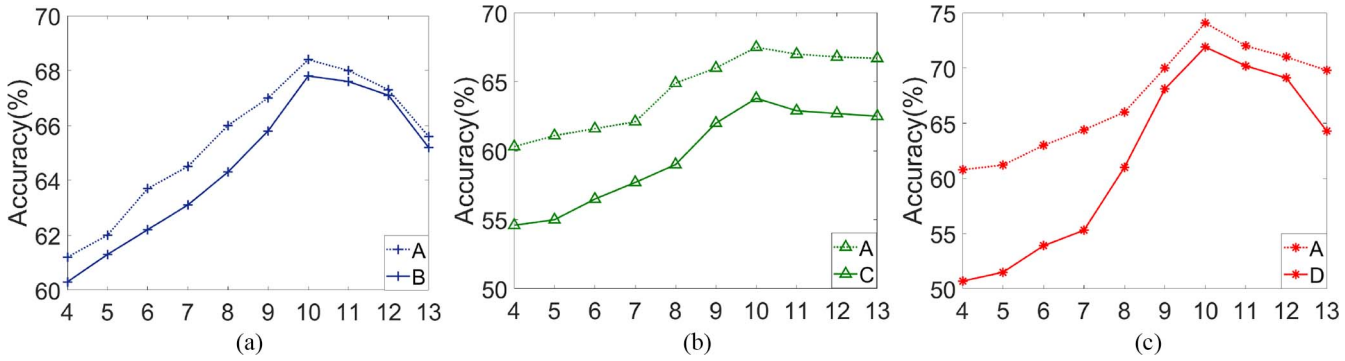


Fig. 11. Parameter sensitivity (k) of DTL on three datasets. (a) $A \rightarrow B$. (b) $A \rightarrow C$. (c) $A \rightarrow D$.

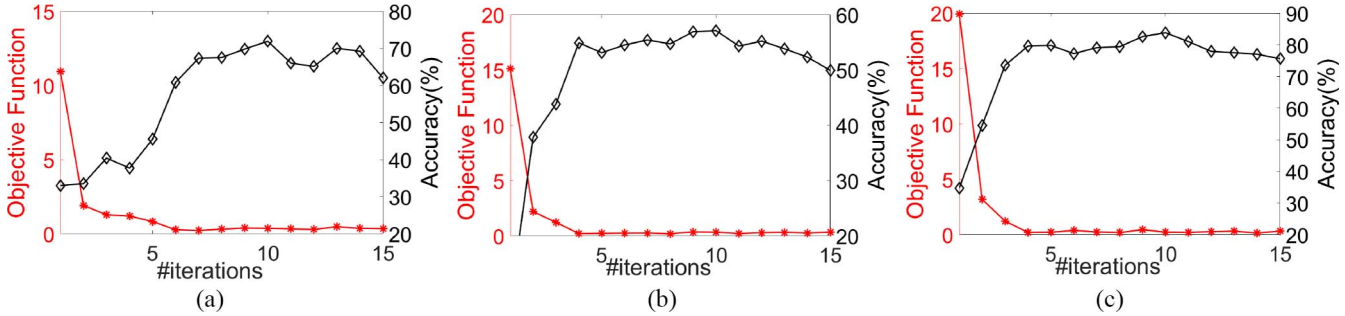


Fig. 12. Convergence study of DTL on three datasets. (a) $A \rightarrow D$. (b) $D \rightarrow B$. (c) $ACDE \rightarrow B$.

TABLE IV
RUNNING TIME (S) AND ACCURACY (%) OF DTL AND COMPARISON TRANSFER LEARNING METHODS ON THREE DATASETS

	Dataset					
	$A \rightarrow D$		$D \rightarrow B$		$ACDE \rightarrow B$	
Method	Runtime	Accuracy	Runtime	Accuracy	Runtime	Accuracy
TCA	5.9	44.1	5.4	44.0	16.4	74.9
GFK	3.1	19.5	2.4	31.6	12.1	57.0
JDA	5092.3	49.3	5078.0	47.7	20082.2	81.9
TJM	3233.5	44.9	3203.0	47.3	12043.6	81.8
ARTL	1565.9	58.0	1562.4	42.8	6102.6	66.8
CORAL	156.9	33.3	151.7	37.1	631.3	68.6
LSA	2.5	39.7	1.9	37.9	8.5	66.7
JGSA	7361.1	55.9	7343.3	51.1	28336.9	79.3
BDA	2649.3	61.8	2639.7	42.8	12031.3	76.3
STL	192.7	30.8	187.4	46.8	870.2	78.5
MEDA	5878.2	43.3	5859.0	43.9	25253.8	67.8
DANN	1659.3	35.1	1650.0	43.2	7653.6	66.6
MSTN	1780.4	39.9	1893.6	42.9	6777.4	64.7
JDDA	408.6	38.1	417.6	48.2	620.8	62.6
DTL	5192.1	71.9	5180.8	57.1	20132.3	83.8

performance is not satisfactory. In contrast, JDA, TJM, JGSA, and BDA exhibit higher performance than the other comparison methods, but they are slow. A predominant computational cost in JDA, TJM, JGSA, BDA, and the proposed DTL is eigendecomposition, whose respective complexity is $\mathcal{O}(n^3)$, $\mathcal{O}(n^3)$, $\mathcal{O}(2n^3)$, $\mathcal{O}(m^3)$, and $\mathcal{O}(m^3)$. In the experiment datasets, m is much smaller than n, and hence the computational time for DTL eigendecomposition is short. Although DTL is time consuming, it retrieves better results than the comparison methods.

5) Learning Tasks: The classification results of DTL, the three conventional classifiers individually (1-NN, SVM, and RF), and the 12 transfer learning methods on the 19 learning tasks are listed in Table V. DTL achieves substantially higher performance than the other methods. The average classification accuracy of DTL is 64.8%, representing a performance improvement of 8.7% compared to the best evaluated method.

The performance of the 1-NN, SVM, and RF classifiers is not satisfactory, because we cannot directly apply them from the source to the target domain. TCA, GFK, TJM, CORAL, and LSA neglect the intraclass affinity, thus undermining performance. Although JDA, ARTL, BDA, and MEDA use pseudolabels for the target data, their initialization has a great influence on the results. JGSA does not perform well due to the two subspaces. The majority voting among classifiers can also improve performance. As the pseudolabel is not iteratively updated via majority voting after the first round, STL cannot achieve high performance. In contrast, DTL refines

TABLE V
CLASSIFICATION ACCURACY (%) OF DRIVING PATTERN RECOGNITION ON PARKING LOT DATASETS

Task	1-NN	SVM	RF	TCA	GFK	JDA	TJM	ARTL	CORAL	LSA	JGSA	BDA	STL	MEDA	DANN	MSTN	JDDA	DTL
$A \rightarrow B(1)$	52.2	58.8	40.5	55.4	50.4	64.9	50.0	50.3	44.4	45.4	60.5	59.4	64.2	47.8	49.5	44.2	46.2	**67.8**
$A \rightarrow C(2)$	29.2	43.2	45.6	47.9	28.4	54.7	52.9	53.5	34.2	44.4	56.2	60.5	40.8	41.2	35.3	44.3	43.8	**63.8**
$A \rightarrow D(3)$	20.2	27.9	29.9	44.1	19.5	49.3	44.9	58.0	33.3	39.7	55.9	61.8	30.8	43.3	35.1	39.9	38.1	**71.9**
$A \rightarrow E(4)$	50.44	58.4	58.0	59.6	45.1	74.5	70.4	53.3	50.2	52.0	73.2	70.4	63.6	57.9	57.0	50.1	55.9	**75.0**
$B \rightarrow A(5)$	49.2	54.4	39.5	54.1	42.8	53.0	49.2	54.5	50.3	47.0	43.3	57.3	56.2	45.2	53.8	47.2	54.6	**61.0**
$B \rightarrow C(6)$	56.0	64.4	56.5	65.1	49.8	62.7	59.7	50.3	51.2	55.8	60.3	61.8	60.4	59.3	50.4	50.4	56.8	**68.1**
$B \rightarrow D(7)$	39.4	41.5	43.2	43.8	31.8	54.9	49.0	46.0	23.3	30.1	55.3	56.4	46.6	39.9	37.8	41.7	45.2	**64.4**
$B \rightarrow E(8)$	30.8	32.5	19.5	41.0	26.3	46.1	33.7	47.6	23.0	21.1	**52.6**	50.5	39.3	35.4	30.8	37.1	32.1	51.7
$C \rightarrow A(9)$	25.1	32.7	28.5	41.8	19.3	54.1	58.1	54.0	38.9	41.0	**60.7**	59.6	35.5	37.7	39.2	43.9	46.5	60.5
$C \rightarrow B(10)$	49.6	64.6	40.8	59.2	47.8	61.8	56.6	50.0	50.5	55.8	60.6	60.2	55.9	55.0	50.6	53.8	52.8	**69.1**
$C \rightarrow D(11)$	44.2	45.5	40.3	49.7	39.0	49.4	50.0	50.1	33.9	41.6	58.2	55.8	46.9	50.4	54.3	46.1	53.0	**68.7**
$C \rightarrow E(12)$	19.2	38.3	22.1	44.2	19.0	50.6	37.5	48.9	35.1	35.0	47.3	53.8	25.1	39.6	26.7	36.3	47.5	**55.6**
$D \rightarrow A(13)$	21.6	40.0	54.8	45.7	19.8	48.9	50.1	40.4	42.6	51.6	53.5	46.1	42.6	51.0	42.8	52.4	46.0	**60.8**
$D \rightarrow B(14)$	36.7	40.1	52.2	44.0	31.6	47.7	47.3	42.8	37.1	37.9	51.1	42.8	46.8	43.9	43.2	42.9	48.2	**57.1**
$D \rightarrow C(15)$	43.4	46.2	46.7	48.7	42.0	53.6	57.3	49.3	44.0	50.2	50.1	59.7	45.7	56.4	47.2	50.7	48.0	**61.8**
$D \rightarrow E(16)$	37.1	47.2	47.7	53.1	34.2	60.8	62.4	65.3	39.8	50.2	56.0	62.8	46.6	43.0	47.5	53.3	49.4	**65.9**
$ACDE \rightarrow B(17)$	68.1	73.3	82.8	74.9	57.0	81.9	81.8	66.8	68.6	66.7	79.3	76.3	78.5	67.8	66.6	64.7	62.6	**83.8**
$ABDE \rightarrow C(18)$	35.0	41.1	39.9	60.4	25.4	69.4	68.3	60.5	56.4	58.0	62.8	69.8	40.2	44.0	40.7	55.9	61.0	**72.5**
$ABCE \rightarrow D(19)$	26.0	33.3	29.6	47.6	17.2	58.6	54.8	35.4	58.9	54.2	51.0	67.4	33.9	37.4	33.1	54.2	58.4	**76.5**
Average	38.6	46.5	43.0	51.6	34.0	57.8	54.4	54.4	42.9	46.0	57.3	59.6	47.0	47.2	44.3	47.8	49.8	**64.8**

TABLE VI
CLASSIFICATION ACCURACY (%) OF DRIVING PATTERN RECOGNITION ON PARKING LOT DATASETS FOR DTL WITH DIFFERENT CLASSIFIERS

Task	DTL-only 1-NN	DTL-only SVM	DTL-only RF	DTL
$A \rightarrow B(1)$	59.3	63.8	62.8	**67.8**
$A \rightarrow C(2)$	55.3	59.8	58.8	**63.8**
$A \rightarrow D(3)$	65.9	65.1	64.0	**71.9**
$A \rightarrow E(4)$	69.0	71.5	70.0	**75.0**
$B \rightarrow A(5)$	57.0	57.5	57.1	**61.0**
$B \rightarrow C(6)$	60.1	67.1	64.1	**68.1**
$B \rightarrow D(7)$	58.8	61.4	60.4	**64.4**
$B \rightarrow E(8)$	47.7	50.7	46.3	**51.7**
$C \rightarrow A(9)$	55.9	56.6	56.5	**60.5**
$C \rightarrow B(10)$	64.7	65.1	64.9	**69.1**
$C \rightarrow D(11)$	60.7	64.1	63.5	**68.7**
$C \rightarrow E(12)$	45.8	51.5	51.6	**55.6**
$D \rightarrow A(13)$	53.2	56.2	54.8	**60.8**
$D \rightarrow B(14)$	51.8	52.8	51.3	**57.1**
$D \rightarrow C(15)$	56.2	56.7	57.0	**61.8**
$D \rightarrow E(16)$	58.0	62.7	61.2	**65.9**
$ACDE \rightarrow B(17)$	75.8	80.8	78.9	**83.8**
$ABDE \rightarrow C(18)$	65.5	69.5	68.5	**72.5**
$ABCE \rightarrow D(19)$	72.3	72.5	71.5	**76.5**
Average	59.6	62.4	61.2	**64.8**

pseudolabels and achieves much better results. Hence, DTL can construct a more effective and robust representation for cross-domain driving pattern recognition.

We also observed that the performance of some methods such as DANN becomes worse when using multiple source domains (*ABCE* $\rightarrow$ *D*) together than single source domain *A* $\rightarrow$ *D*. The degradation might be caused by the distinct data distributions among different domains, since the larger distribution discrepancy is, the worse knowledge transfer efficacy should be expected in transfer learning. DANN only considers the MMD between the source domains (*A*, *B*, *C*, and *E*) and target domain (*D*), hence the distributions in source domains are not aligned. Compared with single source domain, multiple source domains may bring confusion which leads to worse performance. While for our model, the labels available in source domains and pseudolabels in the target domain are used to well align all the different domains, which seems beneficial in knowledge transferring, as demonstrated by the experimental results.

6) Effectiveness of Majority Voting Strategy: We also evaluated the effectiveness of the majority voting strategy by employing weighted 1-NN, SVM, and RF as candidate classifiers. When multiple classifiers have different classification results, the ensemble approach failed to get a unified label. In this article, we simply adopted the result of the SVM method as the final result due to its superior performance on the parking lot dataset. Table VI lists the classification accuracy obtained by integrating the three classifiers using majority voting and that obtained from each individual classifier after feature transformation.

Compared with the results of 1-NN, SVM, and RF in Table V, the performance improves after feature transformation. Hence, feature transformation effectively narrows the gap between domains. After feature transformation, using SVM alone in the proposed DTL outperforms the use of either 1-NN or RF on average. Moreover, the integrated classifiers consistently achieve better results than any of the three classifiers independently in all tested scenarios, demonstrating the effectiveness of the majority voting strategy.

VI. Conclusion

In this article, we have proposed a driving pattern recognition method based on DTL. It aims to adjust intraclass and interclass distributions through an iterative ensemble procedure to enhance the compactness (i.e., reduce intraclass discrepancy) and separability (i.e., increase interclass distance) of transfer learning. Extensive experiments show that DTL is effective and robust for driving pattern recognition and can significantly outperform various state-of-the-art methods on parking lot datasets. We mainly focused on the transfer learning framework for driving pattern recognition in this article. More effective ensemble methods can be considered to integrate classification results from different classifiers and further improve the recognition accuracy, which will be one of our future works.

References

[1] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 62–70, Jan. 2019.

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

[3] L. Zhang, "Transfer adaptation learning: A decade survey," 2019. [Online]. Available: arXiv:1903.04687.

[4] 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.

[5] 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.*, 2014, pp. 1410–1417.

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

[7] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

[8] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban, "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 56–63.

[9] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. Pac. Rim Int. Conf. Artif. Intell.*, 2014, pp. 898–904.

[10] Y. Lin *et al.*, "Cross-domain recognition by identifying joint subspaces of source domain and target domain," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1090–1101, Mar. 2017.

[11] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, "Separability and compactness network for image recognition and superresolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 3275–3286, Nov. 2019.

[12] 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.*, 2013, pp. 2200–2207.

[13] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. IEEE Int. Conf. Data Min.*, 2017, pp. 1129–1134.

[14] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[15] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. ACM Multimedia Conf.*, 2018, pp. 402–410.

[16] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1859–1867.

[17] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, "Stratified transfer learning for cross-domain activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2018, pp. 1–10.

[18] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1541–1546.

[19] F. Z. Zhuang, P. Luo, and C. Y. Du, "Triplex transfer learning: Exploiting both shared and distinct concepts for text classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1191–1203, Jul. 2014.

[20] L. X. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 711–718.

[21] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1785–1792.

[22] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. AAAI Conf. Artif. Intell.*, 2011, pp. 1304–1309.

[23] V. K. Kurmi *et al.*, "Curriculum based dropout discriminator for domain adaptation," 2019. [Online]. Available: arXiv:1907.10628.

[24] M. Harel and S. Mannor, "Learning from multiple outlooks," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[25] M. Gjoreski, S. Kalabakov, M. Lustrek, and H. Gjoreski, "Cross-dataset deep transfer learning for activity recognition," in *Proc. ACM Int. Symp. Wearable Comput.*, 2019, pp. 714–718.

[26] J. Nam, W. Fu, S. Kim, T. Menzies, and L. Tan, "Heterogeneous defect prediction," *IEEE Trans. Softw. Eng.*, vol. 44, no. 9, pp. 874–896, Sep. 2018.

[27] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2010, pp. 1118–1127.

[28] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2213–2219.

[29] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face and kinship verification," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4269–4282, Sep. 2017.

[30] J. Hu, J. Lu, and Y.-P. Tan, "Sharable and individual multi-view metric learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2281–2288, Sep. 2018.

[31] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample problem," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 513–520.

[32] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, "A two-stage weighting framework for multi-source domain adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 505–513.

[33] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[34] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5423–5432.

[35] C. Chen, Z. Chen, B. Jiang, and X. Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3296–3303.

[36] 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.

[37] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 321–328.

[38] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[39] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1601–1608.

[40] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[41] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[42] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

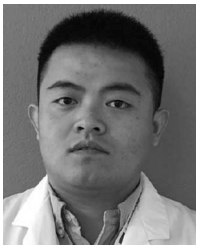
[43] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

Liu Yang (Member, IEEE) received the Ph.D. degree in computer science from the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China, in 2016.

She is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests include data mining and machine learning.

Maoying Li received the master's degree in computer science and technology from the College of Intelligence and Computing, Tianjin University, Tianjin, China, in 2020.

She is currently an Algorithm Engineer with the Automotive Data of China (Tianjin) Company Ltd., Tianjin. Her recent research interests include autonomous driving vision algorithms and machine learning.



Chenyang Shen received the B.Sc. degree from Yangzhou University, Yangzhou, China, in 2010, and the M.Phil. and Ph.D. degrees from Hong Kong Baptist University, Hong Kong, in 2012 and 2015, respectively.

He is an Instructor with the Division of Medical Physics and Engineering, Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX, USA. His current research interests include medical imaging, scientific computing, data mining, and deep learning.

Qinghua Hu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently the Dean of the School of Artificial Intelligence, the Vice Chairman of the Tianjin Branch of China Computer Federation, and the Vice Director of the SIG Granular Computing and Knowledge Discovery, Tianjin University, Tianjin, China. He is currently supported by the Key Program, National Natural Science Foundation of China. He has published over 200 peer-reviewed papers. His current research focuses on uncertainty modeling in big data, machine learning with multimodality data, and intelligent unmanned systems.

Dr. Hu is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Acta Automatica Sinica*, and *Energies*.

Jia Wen is currently pursuing the master's degree in computer science and technology with the College of Intelligence and Computing, Tianjin University, Tianjin, China.

Her research interests include model compression and tensor networks.

Shujie Xu received the B.Sc. degree from Hebei University of Technology, Tianjin, China, in 2009, and the M.Phil. degree from Tianjin University, Tianjin, in 2012.

He is a Senior Engineer with the China Automotive Technology and Research Center Company Ltd., Tianjin. His current research interest is intelligent-connected vehicles.