Personalized Driver Braking Behavior Modeling in the Car-Following Scenario: An Importance-Weight-Based Transfer Learning Approach

Zirui Li, Jianwei Gong, *Member, IEEE*, Chao Lu, *Member, IEEE*, and Jinghang Li

***Abstract*—Accurately recognizing braking intensity levels (BIL) of drivers is important for guaranteeing the safety and avoiding traffic accidents in intelligent transportation systems. In this article, an instance-level transfer learning framework is proposed to recognize BIL for a new driver with insufficient driving data by combining the Gaussian mixture model (GMM) and the importance-weighted least-squares probabilistic classifier (IWLSPC). By considering the statistic distribution, GMM is applied to cluster the data of braking behaviors into three levels with different intensities. With the density ratio calculated by unconstrained least-squares importance fitting, the least-squares probabilistic classifier is modified as IWLSPC to transfer the knowledge from one driver to another and recognize BIL for a new driver with insufficient driving data. Comparative experiments with nontransfer methods indicate that the proposed framework obtains a higher accuracy in recognizing BIL in the car-following scenario, especially when sufficient data are not available.**

***Index Terms*—Braking intensity level (BIL), density ratio estimation, driver model, importance-weighted cross-validation (IWCV), transfer learning (TL).**

I. Introduction

HOW to explicitly and accurately recognize driver's behaviors and intentions during the driving process has been the focus of many studies related to the safety of advanced driver assistance systems (ADAS), automated driving systems, and intelligent transportation systems (ITS) [1]. Recognized results can help to improve the performance of the aforementioned systems and enhance the traffic efficiency, driving safety, and fuel economy [2]. Due to the close relation with driving safety, drivers' braking behaviors have been considered as one of the most important driving behaviors in recent studies [3].

Researchers using both model-based and data-driven approaches contributed a lot to the modeling of braking behaviors, recognition of braking style, identification of braking intensity levels (BIL), and prediction of braking pedal operations. Model-based methods mainly focus on formulating braking behaviors by control theory and parametric approaches. In [4], the speed reduction time was used to describe the braking behavior of drivers. Considering the urban scenario with crossing pedestrians, a parametric model was used to model the drivers' braking behavior. In [5], based on the theory of optimal control, a layered control system was developed to model stopping behaviors (deceleration and braking) by carrying out both action priming and action selection. A new braking system considering the driver's braking intentions and the working characteristics of the motor was proposed in [6], which modeled the relationship between drivers' braking intention and torque to maximize the performance of model-predictive-control-based braking systems.

With the recent development of data science, data-driven methods using machine learning provide a promising way to recognize and predict drivers' behaviors [7]–[10]. For instance, using naturalistic driving data, a Gaussian mixture model (GMM)–hidden Markov model (HMM) method was used in [7] to learn and infer the driver's braking behavior in the car-following scenario. Another data-driven method was proposed in [9], which extracted characteristics and joint features of electroencephalography (EEG) data to judge whether the driver will brake or not. The developed model efficiently distinguished three no-braking emergency situations and presented a satisfactory performance of detection in a wide range of emergency situations. Meanwhile, another EEG-based method was also proposed for brain-controlled vehicles by the application of regularization linear discriminant analysis with spatial-frequency features, which performed a higher accuracy with a smaller time delay [8]. Besides research studies in ADAS, Lv *et al.* [10] proposed a hybrid learning-based approach combining GMM and artificial neural network to classify the driver's BIL and quantitatively infer the braking pressure, which provided an

Manuscript received July 5, 2021; revised November 16, 2021 and January 4, 2022; accepted January 13, 2022. Date of publication February 1, 2022; date of current version May 2, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 61703041 and U19A2083 and in part by the Technological Innovation Program of BIT. (*Corresponding authors: Jianwei Gong; Chao Lu.*)

The authors are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: zirui_li@bit.edu.cn; gongjianwei@bit.edu.cn; chaolu@bit.edu.cn; 3120180346@bit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2022.3146549.
Digital Object Identifier 10.1109/TIE.2022.3146549

additional redundancy for the braking system in the electric vehicle.

Compared with model-based approaches that rely on specific models, data-driven approaches have obvious advantages in the improvement of performance when effective models are not available [11]. Because of the nonlinearity and uncertainty of driver behaviors, effective physical models for drivers cannot be obtained easily. This issue will get even worse when the personalized driver-specific model is required for each individual driver [12]. Under such circumstances, learning-based methods without considering physical models have attracted increasing attention and have been the core issue of many recent studies [11]. One assumption underlying these studies is that sufficient data can be obtained for all drivers. However, this assumption cannot be satisfied in many practical situations, especially when a new driver is being considered. To guarantee the performance of traditional learning-based methods, sufficient data for the new driver are needed and the process of data collection has to be repeated. Collecting sufficient data for the new driver is time-consuming work and needs extra financial support. Therefore, how to efficiently recognize behaviors of newly involved drivers with insufficient driving data is the key to develop effective learning-based methods.

Previous studies proposed to combine sufficient historical data and insufficient new data in one training dataset, but individual characteristics of the new driver cannot be fully described and the demand for new data is still at a high level. Some research studies proposed to solve the problem by parallel learning, which presents a promising approach to formulate the transfer learning (TL) in the application of autonomous driving [13]. A novel learning-based framework is proposed in this article to deal with the problem of driver-specific braking behavior recognition (DSBR) in the car-following scenario. This framework is developed based on the TL technology, which can help to construct an effective recognition system for the new target driver with insufficient driving data available by making full use of historical data collected from other drivers. The main contributions of this article are summarized as follows.

1) A novel TL-based framework, DSBR, is proposed to effectively recognize new driver's BIL with insufficient driving data, which leads to a low-cost and data-efficient approach for driver behavior modeling.
2) A scheme combining supervised and unsupervised methods is designed to automatically label and recognize BIL, which is a hybrid learning framework for transferable driver's behavior modeling.
3) Instead of focusing on whether the driver will brake or not [7], this research study clusters BIL into three categories from the view of the statistic distribution, which provides a more rational analysis of BIL.

The rest of this article is organized as follows. The detailed methodology for the framework is presented in Section II. Section III shows the problem formulation of the driver's braking intensity model and TL. Section IV describes experimental settings, data collection, preprocessing, and finally, analyses of experimental results. Finally, Section V concludes this article.

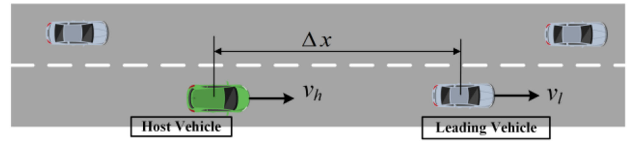


Fig. 1. Car-following scenario considered in this article.

II. Driver-Specific Braking Behavior Recognition

Based on TL, the framework for DSBR is detailed in this section. To formulate an effective TL problem for DSBR, the car-following scenario is considered. Details of the car-following scenario and the proposed DSBR framework are described as follows.

A. Description of the Car-Following Scenario

The car-following scenario is one of the fundamental testing scenarios for modeling and analysis of driver behaviors, which has been used to capture and model the braking behavior of drivers [7]. As shown in Fig. 1, in this article, a typical car-following scenario with one host vehicle and one leading vehicle is selected. The goal of the host vehicle is to follow the leading vehicle by properly accelerating or braking behaviors according to the preference of drivers in the host vehicle. Similar to [10], in our work, the braking behavior is described as the braking intensity that can be divided into several different levels according to the distribution of driving data. The objective of the proposed framework is to successfully recognize driver-specific BIL during the process of car following.

Following [7], the longitudinal behavior of drivers in the car-following scenario is described by three variables: $v_{\mathrm{h},t}$, Δx_t, and $v_{\mathrm{l},t}$. Δx_t is the relative distance between the host and leading vehicles. $v_{\mathrm{h},t}$ and $v_{\mathrm{l},t}$ are velocities of the host and leading vehicles, respectively. Considering the car-following scenario shown in Fig. 1, states of scenario at time t is defined as $\mathbf{s}_t = [v_{\mathrm{h},t} \;\; \Delta x_t \;\; v_{\mathrm{l},t}]^{\mathrm{T}}$. To follow the leading vehicle, drivers in the host vehicle usually perform braking behaviors according to a combined influence of $v_{\mathrm{h},t}$, Δx_t, and $v_{\mathrm{l},t}$ [7], [14]–[16]. Different drivers will perform different brake actions based on the driving experience, gender, and drivers' own judgments of danger. For instance, the expert driver always considers it as a "safety" situation with a higher v_{h} and smaller Δx , whereas the same condition may be affirmed as a "danger" situation by novice.

B. System Framework for DSBR

The proposed DSBR framework is shown in Fig. 2. For a new driver B, different from traditional learning-based methods that need sufficient driving data collected from driver B, the proposed framework tends to use insufficient driving data collected from driver B and sufficient driving data collected from driver A based on the relationship between drivers A and B, which is described by the ratio of density. It can be widely applied in kernel-based methods that focus on applying distribution and covariance function to predict or recognize driver behaviors.

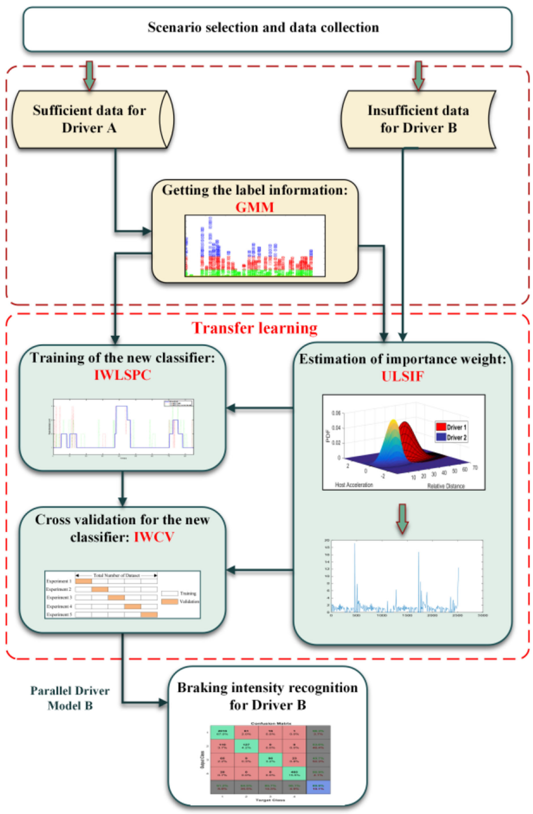


Fig. 2. Specific proposed framework: DSBR.

The proposed framework is composed of four main components, i.e., GMM, unconstrained least-squares importance fitting (ULSIF), importance-weighted least-squares probabilistic classifier (IWLSPC), and importance-weighted cross-validation (IWCV). Considering that different drivers have different thresholds in decision-making of BIL, GMM, an unsupervised method, is used to cluster driving data and automatically label BIL. ULSIF is applied to estimate importance weight (IW). IWLSPC is employed to realize the covariate shift adaptation for DSBR from driver A to driver B, and IWCV is designed for model selection.

In most conditions, rule-based labeling methods have poor ability in the adaption for road situations with different braking demands and drivers with different driving experiences [10]. Therefore, GMM, as a representative unsupervised method, is introduced to classify and obtain the driver's BIL [7], [10]. The GMM is chosen because it has demonstrated its powerful effectiveness in modeling other driving tasks and the stochastic features of driver behavior. The first advantage of GMM is that no labeled data are required in the process of model training, and the number of clusters can be chosen flexibly. The second advantage is that GMM can obtain the probability distribution of braking pressure with an unsupervised learning-based approach. Multivariate GMMs can be represented as follows:

$$G(\mathbf{p}_\mathrm{b}, \theta) = \sum_{i=1}^{K} \pi_i \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_i|^{1/2}} \times \exp\left[-\frac{1}{2}(\mathbf{p}_\mathrm{b} - \mu_i)^\mathrm{T}\mathbf{\Sigma}_i^{-1}(\mathbf{p}_\mathrm{b} - \mu_i)\right] \quad (1)$$

where $\theta = \{\theta\}_{i=1}^{K}$ with $\theta_i = \{\pi_i, \mu_i, \mathbf{\Sigma}_i\}$ and $\mathbf{p}_\mathrm{b} = \{p_{\mathrm{b},i}\}_{i=1}^{K}$ is the braking pressure. μ_i and Σ_i are mean and covariance vectors of the ith single Gaussian model, respectively. π_i is the prior probability with $\sum_{i=1}^{K} \pi_i = 1$. The likelihood function is formulated as follows:

$$p(\mathbf{p}_\mathrm{b}|\theta) = \sum_{i=1}^{K} p(p_{\mathrm{b},i}|\theta). \quad (2)$$

The maximum iteration step is set in advance to obtain estimated optimal parameters until values of the likelihood function meet the maximum or convergent. The objective function to calculate optimal parameters is shown as follows:

$$\hat{\theta} = \arg\max_{\theta} L(\theta) = \arg\max_{\theta} \log(p(\mathbf{p}_\mathrm{b}; \theta)). \quad (3)$$

However, the nonlinearity of (3) with regard to limits to search the optimal value by directly solving (3). Fortunately, the expectation–maximum (EM) algorithm provides a possible means to get the optimal value that maximizes $L(\theta)$ with iteration. Meanwhile, for the first stage of GMM, the K-means algorithm is applied as the initialization step. After the initialization, the soft clustering of GMM with EM algorithm starts to generate the cluster result with probability. The number of clusters is preset. The cluster label a(braking intensity level), as the ground truth of recognized results and the output of GMM, can be obtained by the following function:

$$a = \arg\max_{1\leq l\leq N}\{\Pr(l|p_{\mathrm{b},i})\}. \quad (4)$$

III. Problem Formulation for TL

In order to realize the target of successfully recognizing the target driver's BIL by TL from source driver's driving data, the following methods are combined and developed as the DSBR framework: ULSIF, IWLSPC, and IWCV. The overall architecture of DSBR is shown in Fig. 2.

A. Significance of TL

To build a driver's braking intensity model with better performance, as many as possible situations should be covered as training data. But the collection and analysis of personalized driving data have following limitations.

1) The collection of sensor-based driving data for naturalistic driving data is expensive. Due to complicated traffic conditions in the real road, such as traffic jams, the useless and useful driving data are mixed as a whole. Therefore, the process of analysis is always time-consuming, which includes manually extracting situations, model training, and testing.
2) Driving data collected in the simulation have a lower cost compared to the collection of naturalistic driving data. However, the simulated platform cannot fully reflect complicated and real road conditions.
3) The driver model built by personalized driving data collection and analysis only fits the single driver whose driving data are collected and analyzed [17], [18]. The performance of the model decreases rapidly for target driver's driving data. Therefore, for a new driver with few

driving data, the personalized driver model has limited ability in the adaptation and generalization.

Three limitations mentioned above for personalized data collection restrict the adaptation of the model between two drivers. Therefore, TL is introduced as a new way to transfer the knowledge and parameters of the model from the source driver (sufficient driving data) to the target driver (insufficient driving data).

It is worth noting that if all drivers' driving data are sufficient and fully supplied, TL is not required. In this article, the proposed method mainly considers the situation where sufficient driving data are collected from the source driver while insufficient data are supplied for the target driver. Datasets of source and target drivers are named as source domain $\mathbf{D}_{\mathrm{S}}$ and target domain $\mathbf{D}_{\mathrm{T}}$, respectively.

Most instance-transfer approaches to the transductive TL setting are motivated by importance sampling. In general, we want to learn optimal parameters θ^* by minimizing the expected risk [19], [20]

$$\theta^* = \arg\min_{\theta\in\Theta} \mathrm{E}_{(x,y)\in P}[l(x,y,\theta)] \tag{5}$$

where $l(x,y,\theta)$ is a loss function that depends on the parameter θ. However, since it is hard to estimate the probability distribution P, we choose to minimize the empirical risk minimization

$$\theta^* = \arg\min_{\theta\in\Theta} \sum_{\zeta\in\mathbf{D}_{\mathrm{T}}} P(\mathbf{D}_{\mathrm{T}}) l(\zeta_t,\theta) \tag{6}$$

where $\zeta_t = [\mathbf{s}_t, a_t]$ and $l(\zeta_t,\theta)$ is the loss function. The original hypothesis for the proposed framework is that distributions of source and target driving data are different, $P(\mathbf{D}_{\mathrm{S}}) \neq P(\mathbf{D}_{\mathrm{T}})$. Since insufficient labeled BIL are given in the target domain as training data, we cannot obtain the target model with a high accuracy in the recognition. The DSBR framework proposes to learn the target model from the training data of the source domain. Therefore, the risk function can be described as follows:

$$\begin{aligned}\theta^* &= \arg\min_{\theta\in\Theta} \sum_{\zeta\in\mathbf{D}_{\mathrm{T}}} \frac{P(\mathbf{D}_{\mathrm{T}})}{P(\mathbf{D}_{\mathrm{S}})} P(\mathbf{D}_{\mathrm{S}}) l(\zeta_{\mathrm{T}},\theta)\\ &\approx \arg\min_{\theta\in\Theta} \sum_{i=1}^{n_{\mathrm{s}}} \frac{P_{\mathrm{T}}(\zeta_{\mathrm{T},i})}{P_{\mathrm{S}}(\zeta_{\mathrm{S},i})} P_{\mathrm{T}}(\zeta_{\mathrm{T},i}) l(\zeta_{\mathrm{S},i},\theta).\end{aligned} \tag{7}$$

According to [20, Definition 1], for the transductive TL with the same learning task in source and target domains, we have $P(\mathbf{a}_{\mathrm{T}}|\mathbf{s}_{\mathrm{T}}) = P(\mathbf{a}_{\mathrm{S}}|\mathbf{s}_{\mathrm{S}})$. Therefore, we can get the following equation:

$$\frac{P_{\mathrm{T}}(\mathbf{s}_{\mathrm{T},i}, a_{\mathrm{T},i})}{P_{\mathrm{S}}(\mathbf{s}_{\mathrm{S},i}, a_{\mathrm{S},i})} = \frac{P_{\mathrm{T}}(\mathbf{s}_{\mathrm{T},i})}{P_{\mathrm{S}}(\mathbf{s}_{\mathrm{S},i})}. \tag{8}$$

Therefore, the final task to build a target model is to accurately estimate or infer $P_{\mathrm{T}}(\mathbf{s}_{\mathrm{T},i})/P_{\mathrm{S}}(\mathbf{s}_{\mathrm{S},i})$, which is solved by ULSIF.

B. Importance Estimator: ULSIF

The key point of instance-level TL is to estimate the ratio $P_{\mathrm{T}}(\mathbf{s}_{\mathrm{T},i})/P_{\mathrm{S}}(\mathbf{s}_{\mathrm{S},i})$ between two different density functions. The ratio can be used for some applications, such as covariate shift adaptation [19]. The density ratio estimator is developed based on the assumption that distributions of the source and target domains are independent and identically distributed. ULSIF is a typical method to estimate IW by transforming the importance fitting into the least-square (LS) problem. It can be transferred into a convex problem, which can be solved by a standard quadratic program solver.

To estimate $\omega(s) = p_{\mathrm{T}}(\mathbf{s}_{\mathrm{T}})/p_{\mathrm{S}}(\mathbf{s}_{\mathrm{S}})$, the linear model is used to describe $\omega(s)$. The Gaussian kernel function is selected as the basic function and $\mathbf{s}_{\mathrm{T}} = \{\mathbf{s}_{\mathrm{T},i}\}_{i=1}^{N_{\mathrm{T}}}$ are chosen as the Gaussian kernel center.

$$\hat{\omega}(\mathbf{s};\alpha) = \Sigma_{l=1}^{b} \alpha_l K_\sigma(\mathbf{s}, \mathbf{s}_{\mathrm{T},l}) \tag{9}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_b)^{\mathrm{T}}$ are parameters of the model, which need to be learned by the training process. b is the number of selected samples in the target domain, which is applied to estimate $\omega(s)$. $K_\sigma(\mathbf{s}, \mathbf{s}_{\mathrm{T},l})$ is the Gaussian kernel with a kernel width σ. α is determined by minimizing the following LS function:

$$\begin{aligned}J(\alpha) &= \frac{1}{2}\int (\omega(\mathbf{s};\alpha) - \omega(\mathbf{s}))^2 p_{\mathrm{T}}(\mathbf{s})\mathrm{d}\mathbf{s}\\ &= \frac{1}{2}\int \omega(\mathbf{s};\alpha)^2 p_{\mathrm{S}}(\mathbf{s})\mathrm{d}\mathbf{s} - \int \omega(\mathbf{s};\alpha)^2 p_{\mathrm{T}}(\mathbf{s})\mathrm{d}\mathbf{s} + C'\\ &= \frac{1}{2}\alpha^T \mathbf{H}\alpha - \mathbf{h}^T\alpha + C'\end{aligned} \tag{10}$$

where C' is a constant, $\mathbf{H}$is $N_{\mathrm{T}} \times N_{\mathrm{T}}$ matrix, and $\mathbf{h}$ is the vector with N_{T} dimension, which are defined as follows:

$$H_{n,n'} = \int K_\sigma(\mathbf{s}, \mathbf{s}_{\mathrm{T},n}) K_\sigma(\mathbf{s}, \mathbf{s}_{\mathrm{T},n'}) p_{\mathrm{S}}(\mathbf{s})\mathrm{d}\mathbf{s} \tag{11}$$

$$h_n = \int K_\sigma(\mathbf{s}, \mathbf{s}_{\mathrm{T},n}) p_{\mathrm{T}}(\mathbf{s})\mathrm{d}\mathbf{s} \tag{12}$$

where $\mathbf{H}$ and $\mathbf{h}$ can be approximated by the empirical averages

$$\hat{\mathbf{H}}_{n,n'} = \frac{1}{N_{\mathrm{S}}} \sum_{n''}^{N_{\mathrm{S}}} K(\mathbf{s}_{\mathrm{S},n''}, \mathbf{s}_{\mathrm{T},n}) K(\mathbf{s}_{\mathrm{S},n''}, \mathbf{s}_{\mathrm{T},n}) \tag{13}$$

$$\hat{\mathbf{h}}_n = \frac{1}{N_{\mathrm{T}}} \sum_{n'=1}^{N_{\mathrm{T}}} K(\mathbf{s}_{\mathrm{T},n'}, \mathbf{s}_{\mathrm{T},n}) \tag{14}$$

where $\hat{\mathbf{H}}$ and $\hat{\mathbf{h}}$ are optimal values. The parameter α can be obtained by solving an optimization problem

$$\hat{\alpha} = \arg\min_{\alpha} \left(\frac{1}{2}\alpha^{\mathrm{T}}\mathbf{H}\alpha - \hat{\mathbf{h}}^{\mathbf{T}}\alpha + \frac{\gamma}{2}\alpha^{\mathbf{T}}\alpha\right) \tag{15}$$

where $\frac{\gamma}{2}\alpha^{\mathbf{T}}\alpha$ is the part for regularization. The analytical solution is applied to calculate α

$$\hat{\alpha} = (\hat{\mathbf{H}} + \gamma\mathbf{I}_{N_{\mathrm{te}}})\hat{\mathbf{h}}. \tag{16}$$

Finally, the nonnegative IW $\omega(s)$ is calculated as follows:

$$\hat{\omega}(s) = \max\left(0, \sum_{n=1}^{N_{\mathrm{T}}} \hat{\alpha}_n K_\sigma(\mathbf{s}, \mathbf{s}_{\mathrm{T}})\right). \tag{17}$$

C. Importance-Weighted Least-Squares Probabilistic Classifier

In order to obtain recognized results based on TL, IWLSPC is introduced, which combines the least-squares probabilistic classifier and IW provided by ULSIF [19], [21]. First, a kernel-based model is built to describe and calculate the class-posterior probability

$$p(\mathbf{a}|\mathbf{s};\theta_a)=\sum_{n=1}^{N_\mathrm{T}}\theta_{a,n}K(\mathbf{s},\mathbf{s}_{\mathrm{T},n}) \tag{18}$$

where θ_a is the parametric vector, and $K(\mathbf{s},\mathbf{s}_{\mathrm{T},n})$ is the kernel function. $\theta_{a,n}$ is determined by minimizing the square error

$$\begin{aligned} J(\theta_a) &= \frac{1}{2}\int (p(\mathbf{a}|\mathbf{s};\theta_a)-p(\mathbf{a}|\mathbf{s}))^2 p_\mathrm{T}(\mathbf{s})\mathrm{d}\mathbf{s} \\ &= \frac{1}{2}\int p(\mathbf{a}|\mathbf{s};\theta_a)^2 p_\mathrm{S}(\mathbf{s})\mathrm{d}\mathbf{s} - \int p(\mathbf{a}|\mathbf{s})^2 p_\mathrm{T}(\mathbf{s})\mathrm{d}\mathbf{s} + C \\ &= \frac{1}{2}{\theta_a}^T\mathbf{Q}\theta_a - \mathbf{q}^T\theta_a + C \end{aligned} \tag{19}$$

where C is the constant independent of θ_a. $\theta_{a,n}$ can be calculated by the following equation:

$$\hat{\theta}_a=\underset{\theta_a}{\mathrm{argmin}}\left[\frac{1}{2}\theta_a^\mathrm{T}\hat{\mathbf{Q}}\theta_a - \hat{\mathbf{q}}_a^\mathrm{T}\theta_a + \frac{\lambda}{2}\theta_a^\mathrm{T}\theta_a\right]. \tag{20}$$

In the abovementioned equation, λ is the regularization parameter to avoid overfitting problem. $\hat{\mathbf{Q}}$ and $\hat{\mathbf{q}}_a$ can be approximated based on IW $\omega(s_\mathrm{S})$ by using data from source and target domains, which are similar to the calculation of $\hat{\mathbf{H}}$ and $\hat{\mathbf{h}}$ in (13) and (14)

$$\hat{\mathbf{Q}}_{n,n} = \frac{1}{N_\mathrm{S}}\sum_{n''}^{N_\mathrm{S}} K(\mathbf{s}_{\mathrm{S},n''},\mathbf{s}_{\mathrm{T},n})K(\mathbf{s}_{\mathrm{S},n''},\mathbf{s}_{\mathrm{T},n})\omega(\mathbf{s}_{\mathrm{S},n''})^\upsilon \tag{21}$$

$$\hat{\mathbf{q}}_{a,n} = \frac{1}{N_\mathrm{S}}\sum_{n':\mathbf{a}_{\mathrm{S},n'}=a} K(\mathbf{s}_{\mathrm{S},n'},\mathbf{s}_{\mathrm{T},n})\omega(\mathbf{s}_{\mathrm{S},n'})^\upsilon \tag{22}$$

where the flattening parameter υ is used to controls the bias-variance tradeoff in importance sampling. More specifically, if υ is close to 0, the importance weights are used as they are; then, the bias gets smaller, but the variance tends to be larger. On the other hand, if υ is close to 0, the importance weights tend to be one (i.e., flat). Then, the bias is larger, but the variance is smaller. Consequently, the parameter vector θ_a in (18) can be solved analytically by the following equation:

$$\hat{\theta}_a = (\hat{\mathbf{Q}} + \lambda\mathbf{I}_{N_\mathrm{T}})^{-1}\hat{\mathbf{q}}_a. \tag{23}$$

Finally, the model can learn the class-posterior probability $p(a_\mathrm{T}|\mathbf{s}_\mathrm{T})$. Predicted results are obtained by

$$\hat{\mathbf{a}}_\mathrm{T} = \arg\max(p(\mathbf{a}|\mathbf{s}_\mathrm{T})) \tag{24}$$

where

$$p(a|\mathbf{s}_\mathrm{T}) = \frac{1}{Z}\max\left(0, \sum_{n=1}^{N_\mathrm{T}}\hat{\theta}_{a,n}K_\sigma(\mathbf{s},\mathbf{s}_\mathrm{T})\right). \tag{25}$$

If $Z = \sum_{a=1}^{c}\max(0, \sum_{n=1}^{N_\mathrm{T}}\hat{\theta}_{a,n}K_\sigma(\mathbf{s},\mathbf{s}_\mathrm{T})) > 0$; otherwise $p(a|\mathbf{s}_\mathrm{T}) = \frac{1}{c}$.

D. Importance-Weighted Cross-Validation

The choice of parameters in the proposed framework has a great influence on the recognition of driver's BIL. Each kernel has its own parameters independently. Parameters for different kernels are detailed in Table I. Alternative values for υ, λ, and σ are $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$, $\{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^0\}$, and $\{0.1, 0.2, 0.5, 0.6, 2, 3\}$, respectively. Meanwhile, selected values for polynomial kernel are $\alpha \in \{0.1, 0.3, 0.5, 1\}$, $c \in \{0.1, 0.3, 0.5, 1\}$, and $d \in \{1, 2\}$. Cross validation (CV) is a common and standard method to select parameters of the model. In [21], IWCV is applied and obtain a better performance than the CV-based model selection. In this research, IW is considered in the process of model selection. We randomly divide the training dataset $\mathbf{D}_\mathrm{S}$ into K subsets $\{\mathbf{D}_{\mathrm{S},k}\}_{k=1}^K$. $\hat{f}_k(\mathbf{s}_\mathrm{S})$ is the function trained by dataset $\mathbf{D}_\mathrm{S}\backslash\mathbf{D}_{\mathrm{S},k}$. The generalized error based on IWCV is given as follows:

TABLE I
PARAMETER FOR DIFFERENT KERNELS

Kernel	Expression	Parameters
Gaussian kernel	$K_\sigma(\mathbf{s},\mathbf{s}_\mathrm{T}) = \exp(-\frac{\lVert\mathbf{s}-\mathbf{s}_\mathrm{T}\rVert^2}{2\sigma^2})$	υ , λ , σ
Polynomial kernel	$K(\mathbf{s},\mathbf{s}_\mathrm{T}) = (\alpha\mathbf{s}_\mathrm{T}\mathbf{s}^\mathrm{T}+c)^d$	υ , λ , α , c , d
Linear kernel	$K(\mathbf{s},\mathbf{s}_\mathrm{T}) = \mathbf{s}_\mathrm{T}\mathbf{s}^\mathrm{T}$	υ , λ

$$W_\mathrm{IWCV} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|\mathbf{D}_{S,k}|}\sum_{(\zeta_S\in\mathbf{D}_{S,k})}\omega(\mathbf{s}_S)\mathrm{loss}(\hat{f}_k(\mathbf{s}_S),\mathbf{a}_S) \tag{26}$$

where $\omega(\mathbf{s}_S)$ is IW, which can be calculated by ULSIF. $\mathrm{loss}(\cdot)$ is the loss function to measure the discrepancy or difference between the output $\hat{f}_k(\mathbf{s}_S)$ and ground truth

$$\mathrm{loss} = \begin{cases} 1-\sin(R_\mathrm{true}) \\ 1-\sin(R_\mathrm{false}) \end{cases} \tag{27}$$

where $R_\mathrm{true} = 1$ and $R_\mathrm{false} = -1$ are the rewards for true and false recognition, respectively. Finally, we can get the optimal parameters (example case: Gaussian kernel)

$$(\hat{\lambda}, \hat{\sigma}, \hat{\upsilon}) = \underset{(\lambda,\sigma,\upsilon)}{\arg\min}\, W_\mathrm{IWCV}(\lambda,\sigma,\upsilon). \tag{28}$$

With optimal parameters $(\hat{\lambda}, \hat{\sigma}, \hat{\upsilon})$, recognized results for target driver's BIL can be illustrated as follows:

$$f(\mathbf{s}_{\mathrm{S},1:t}, \mathbf{a}_{\mathrm{S},1:t}, \mathbf{s}_{\mathrm{T},1:t}, s_{\mathrm{T},t+1}; a_{\mathrm{T},t+1})_{(\hat{\lambda},\hat{\sigma},\hat{\upsilon})} : s_{\mathrm{T},t+1} \rightarrow \hat{a}_{\mathrm{T},t+1} \tag{29}$$

where $f(\cdot)$ is the target driver model, which is built by data from source and target drivers. The target driver's state $s_{\mathrm{T},t+1}$ at time $t+1$ is considered as the input to obtain the recognized result $\hat{a}_{\mathrm{T},t+1}$. In the process of evaluation and comparative study, the recognized accuracy of BIL is calculated as the follows:

$$\mathrm{Accuracy} = \frac{N_1+N_2+N_3+N_4}{N} \tag{30}$$

where N is the number of testing samples in the target domain. N_1, N_2, N_3, and $N_4$are the numbers of correct recognition cases, respectively.

IV. Experiments

A. Experimental Setting and Data Collection

The target of the proposed framework is to investigate the performance of TL and model adaptation in the following two conditions: transfer the driver model from virtual to virtual and from naturalistic to naturalistic. Therefore, both virtual and naturalistic driving datasets are required. For comparative study, the balanced distributional adaptation (BDA) is selected [22]. The algorithm is implemented in MATLAB.[1]

1) Simulated Driving Data: To test and evaluate the proposed framework, the PRESCAN simulation platform is applied. To realize the human–vehicle close loop system, Logitech G27 is equipped to collect human drivers' driving data, and three viewing screens are used to provide the view of driving conditions. In the driving simulator, sensors (GPS ans Lidar) are previously set on the simulating vehicle [22]. During experiments, the driver in the host vehicle is asked to follow the leading vehicle in the same lane without cut in/out and overtaking behaviors. In the process of car following, the leading vehicle brakes or decelerates randomly. The driver in the host vehicle needs to follow the leading vehicle according to the change of relative distance between the two vehicles. Following data are collected from the simulator and used for analysis: v_{host}, a_{host}, d_{relative}, v_{relative}, and simulation timestamp $t_{\text{timestamp}}$. Initial velocities of the host and leading vehicles change from 43.2 to 54 km/h and from 36 to 43.2 km/h.

2) Naturalistic Driving Data: Naturalistic driving data collected from the public UAH DriveSet are used to test the performance of the DSBR framework [23]. The public dataset provides 500 min of naturalistic driving data from a smartphone, which include GPS/IMU data, processed semantic information for images, and relative distance for surrounding vehicles. UAH DriveSet consists of six different drivers' driving data with different vehicles (Mercedes, Audi, etc.), two road conditions (motor-way and secondary road), and three different drivers' behaviors (normal, drowsy, and aggressive). In UAH dataset, there is no information from the braking pedal, so the change of acceleration is used to describe driver's BIL [24], [25]. But for a driver's braking action, there exists a delay between the driver's braking action and the change of acceleration. Details of the process that let deceleration replace the information of braking pedal are presented in [24] and [25]. The details of collected data are shown in Table II

B. Data Process

For naturalistic driving dataset and data collected in the simulated environment, useful data and driving conditions for model training and analysis are mixed with useless data. Therefore, before the process of model training, data extraction is required to generate the satisfied driving data for the car-following behavior. As for driving data collected in the simulation, the following conditions are removed from the dataset:

[1]The software toolbox developed in MATLAB by the authors and the collected simulated and naturalistic driving data can be obtained from the first and corresponding authors on reasonable request, which needs the agreement for copyright from the university and the funding organization.

TABLE II
Illustration of the Collected Data

Item	Illustration
GPS	Latitude, longitude, Course, Timestamp.
Accelerometers	Acceleration with Kalman Filter
Process data (vehicle detection)	Distance to the ahead vehicle in the same lane, Time of impact to ahead vehicle, Number of detected vehicles in this frame

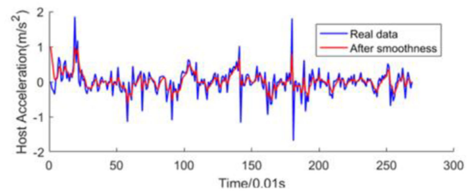


Fig. 3. Illustration of the moving average filter method for host acceleration.

1) collisions between the host vehicle and the leading vehicle;
2) two vehicles are not in the same road lane;
3) the relative distance is larger than 70 m.

As for naturalistic driving data, considering the condition that different kinds of leading vehicles have different influences on the driver's behavior, some specific vehicles (for example, truck) are not chosen as leading vehicles. The following conditions are removed from the dataset:

1) cut in/out of the leading and host vehicles;
2) the time duration is less than 20 s;
3) collisions between the host vehicle and the leading vehicle;
4) two vehicles are not in the same road lane;
5) the relative distance is larger than 70 m.

The moving average filter method is used to smooth raw driving data. The details and comparison of smoothness is shown in Fig. 3.

C. Different Distributions Between Drivers

The original intention of TL is to investigate instance-level knowledge transfer between two drivers with different driving styles in the car-following behavior. Therefore, analyzing the difference of two drivers' behaviors is the first step.

From the view of statistical learning theory, the essential difference between drivers is the different distributions of driving data. For instance, expert drivers tend to keep a small relative distance or gap in the car-following scenario. The distribution of relative distance for expert drivers' driving data concentrates at a smaller value compared to the novice. This article focuses on the joint distribution between host acceleration and relative distance, which can be presented in the form of the multivariate Gaussian regression function as (1) and (2). As shown in Fig. 4, the joint probability distribution of driver 2 has a shorter relative distance and a larger scale of host acceleration compared to driver 1. The distinct difference between driver 1 and driver 2 indicates that drivers 1 and 2 have different joint probability distributions.

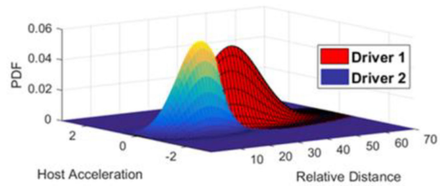


Fig. 4. Comparison of joint distribution between two drivers. Driver 1: the novice driver. Driver 2: the expert driver.

D. Labeling the Braking Intensity Level Based on GMM

For naturalistic driving data, the deceleration of the host vehicle is chosen as the input of GMM. The output of GMM is the identification results of BIL. The number of clusters is selected based on BIC. However, although BIC can investigate optimal GMM components, the index of GMM components and the performance of the driving model are different. More components could result in the overfitting problem, whereas less components could decrease the accuracy of prediction because some characteristics of data cannot be identified, which is supported by the following two theoretical foundations.

1) According to [26], three different levels of deceleration are separately defined: maximum peak deceleration (MPD), maximum deceleration average (MDA), and full effective deceleration (FED), which are applied to describe main braking patterns.
2) In [10], the selection of cluster number K is based on the tradeoff between computation cost and fitting accuracy.

Considering the classification of BIL, $K = 3$ is finally selected as the cluster number. As GMM is specifically applied in identifying brake intensity levels, the tradeoff between proportions of each single Gaussian model and the diversity of clusters are taken into consideration. Finally, BIL are labeled into three clusters: low-intensity braking, middle-intensity braking, and high-intensity braking.

From the two theoretical foundations that have been discussed above, all available driving data collected from the UAH naturalistic dataset are clustered into three groups, except for driving data with zero brake pedal (As for UAH-dataset, deceleration equals zero). Labeled results are shown in Fig. 5(a). Yellow points stand for "no braking," which are selected directly from the available dataset. Three other clusters, as the output of GMM, are named: high-, mid-, and low-intensity braking, respectively. According to the clustering results of GMM, the proportions and range for three clusters are separately shown as follows:

$$\text{Label} = \begin{cases} \text{Low} & 0 < x < 1.1 \\ \text{Middle} & 1.1 < x < 3.3 \\ \text{High} & x > 3.3 \end{cases}. \tag{31}$$

Fig. 5(b) demonstrates the Gaussian distribution of GMM with three components. Samples with a low-intensity braking level are more than samples of the other two levels, so from 0 to 1.1, the value of density for this cluster is higher than the others. And high- and mid-intensity braking clusters are overlapped with each other, which indicates that these two clusters have similar characteristics and cannot be distinguished

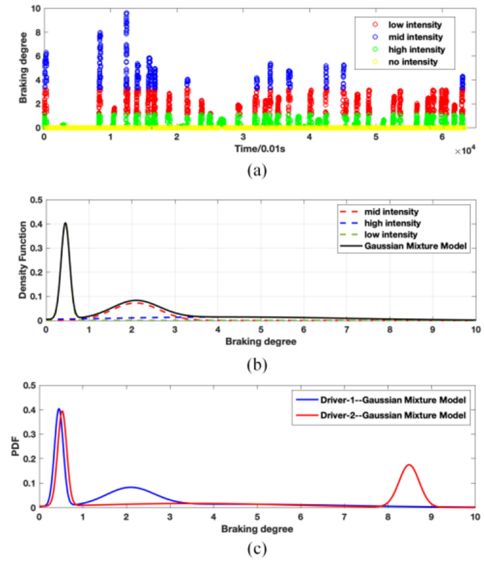


Fig. 5. (a) Clustering results of BIL. (b) Distribution of GMM for BIL. (c) Comparison of Gaussian distribution between different drivers based on GMM.

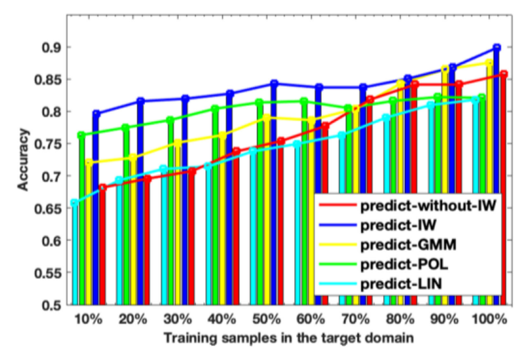


Fig. 6. Comparison of accuracy between different methods with the increase of *training data* (simulated driving data).

thoroughly. As shown in Fig. 5(c), two GMMs are presented for two drivers' driving data. For the low-intensity braking, two drivers' distributions have a slight difference. But for the other part of the two GMMs, the distribution for driver 1 is relatively centered at the range of 1–3, whereas the range of driver 2 is from 8 to 9. Comparative results indicate that driver 2 tends to act "deeper" braking actions than driver 1. It also illustrates the "different distribution" assumptions in this research.

E. Experiment I: Transfer Between Simulated Driving Data

In order to make a comparison between the TL-based method and traditional methods, GMM and LSPC are chosen as baseline methods. Training data for LSPC (radial basis function, RBF) are selected from the source domain, and training data for the other four baseline methods are selected from the target domain. With the increase of *training data*, Fig. 6 presents the change of accuracy in the recognition.

1) General Analysis: As shown in Fig. 6, on the whole, as the number of training samples increases from 120 to 1200

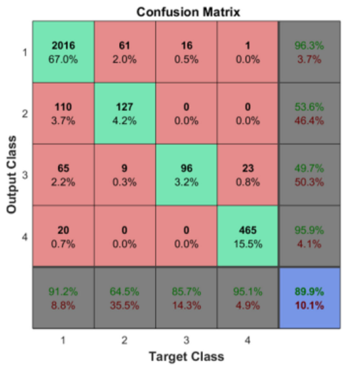


Fig. 7. Confusion matrix of recognized results based on IWLSPC (RBF) with simulated driving data.

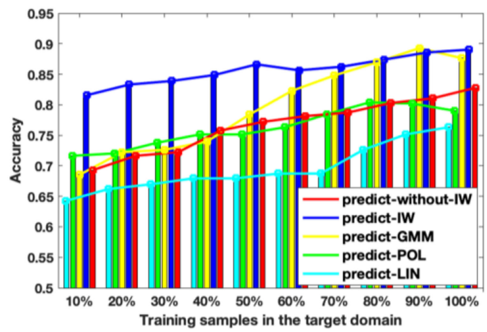


Fig. 8. Comparison of accuracy between different methods with the increase of training data—naturalistic driving data.

(10%–100%), the accuracy of all five methods increases obviously. Comparing IWLSPC (RBF) with GMM, the accuracy of both models increases (in IWLSPC (RBF), it increases from 0.78 to 0.90, and in GMM, it increases from 0.70 to 0.86.). But when training data are 120 samples (10%), the accuracy of IWLSPC (RBF) is 0.08 higher than GMM. It indicates that when the target domain has insufficient driving data for model training, TL-based model IWLSPC performs better with the help of importance weight and the prior knowledge from the source domain. In contrast, the accuracy of GMM is relatively lower (0.70) for the reason that insufficient training data are provided. The lowest gap of accuracy between IWLSPC (RBF) and GMM is 0.003. The decreasing tendency of the gap between IWLSPC (RBF) and GMM indicates that sufficient training data lead to the indiscrimination of traditional methods (without TL) and TL-based methods. For the comparative study with TL-based methods (proposed and BDA), the proposed framework can obtain most of the best performance. The reason may be that the balanced factor in BDA is difficult to select manually in the training process.

2) Comparison Between IWLSPC (RBF) and LSPC (RBF): As for LSPC (RBF), the definition of training data is driving data from the source domain, which is different from the definition of training data for IWLSPC (RBF). With the increase of training data, the recognized accuracy of LSPC (RBF) increases from 0.68 to 0.85. But the highest value of accuracy is 0.85 [IWLSPC (RBF): 0.90; GMM: 0.86]. When the number of training samples increases from 120 to 1200 samples, the gap between IWLSPC (RBF) and LSPC (RBF) is always larger than 0.09. It indicates that insufficient training data in the source domain have a poor performance in the target domain and TL-based method IWLSPC (RBF) can effectively realize the instance-level knowledge transfer. Even with sufficient training data, the highest accuracy for LSPC (RBF) is 0.85, which indicates that ULSIF contributes a more adaptive driver model with a higher accuracy.

3) Comparison Between Different Kernels: IWLSPC is a kernel-based least-squares classification method. RBF is discussed and compared above. In order to investigate the influence of different kernels, linear kernel and polynomial kernel are chosen as baseline kernels for the evaluation of performance. As shown in Fig. 6, RBF performs best among the three kernels. Meanwhile, when the number of training data reaches 1200 samples, the accuracies of linear and polynomial kernels have a tiny difference (0.810 and 0.812). Before the two kernel-based models mentioned above reach their best performance, the recognized accuracy of polynomial kernel is always higher than linear kernel (from 120 to 1080 samples). The accuracy shows an increasing tendency steadily (from 0.748 to 0.81) while linear-kernel-based IWLSPC goes up from a relatively low accuracy (0.658). The detailed performance of IWLSPC (RBF) is shown in Fig. 7. The optimal parameters of this case study are $\upsilon = 0.2$, $\lambda = 10^{-0.5}$, and $\sigma = 0.5$. A total of 3009 data points in the target domain are labeled by GMM as ground truth, which include 2211 no braking, 197 low braking intensity, 112 middle braking intensity, and 489 high braking intensity. In the confusion matrix, the accuracies of recognizing the above-mentioned braking intensities are 91.2%, 64.5%, 85.7%, and 95.1%, respectively, whereas the overall accuracy is 89.1%. In order to fully verify the DSBR framework, besides the experiment introduced above (from driver 1 to 2), five other experiments are also operated: driver 2 to 1, driver 1 to 3, driver 3 to 1, driver 2 to 3, driver 3 to 2. Details of the experiments are shown in Table III.

F. Experiment II: Transfer Between Naturalistic Driving Data (Motorway and Secondary Roads)

The target of the DSBR framework is to realize the instance-level knowledge transfer using both simulated and naturalistic driving data. Therefore, after the comparative study for simulated driving data above, a similar study for naturalistic driving data is also presented. On the whole, according to Fig. 8, as for the comparison of accuracy between different methods with the increase of *training data*, the increasing tendency of accuracy is the same as simulated driving data. An 80-s continuous car-following behavior in the UAH dataset is extracted as ground truth. Four braking actions happen in the 80-s car following. Compared to four baseline methods, IWLSPC (RBF) performs better in the first and third braking actions and most of no braking conditions. As for the confusion matrix of IWLSPC (RBF), accuracies of recognition are 92.7%, 76.3%, 80.0%, and 88.1%, whereas the overall accuracy is 89.0%, which is slightly lower than simulated driving data. The optimal parameters for Fig. 9 are $\upsilon = 0.6$, $\lambda = 10^{-1}$, and $\sigma = 0.2$. To validate the influence of parameter tuning, the flattening parameter is changed from

TABLE III
EXPERIMENTAL RESULTS FOR SIMULATED DRIVING DATA (EXPERIMENTAL I)

Number of Training Data	120 samples (10%)	240 samples (20%)	360 samples (30%)	480 samples (40%)	600 samples (50%)	720 samples (60%)	840 samples (70%)	960 samples (80%)	1080 samples (90%)	1200 samples (100%)
IWLSPC (3 to 1)	**0.778**	**0.801**	**0.818**	**0.829**	**0.841**	**0.853**	0.865	**0.867**	**0.877**	0.882
BDA (3 to 1)	0.768	0.792	0.780	0.812	0.833	0.850	**0.877**	0.852	0.864	0.878
IWLSPC (2 to 1)	0.771	0.781	0.798	0.822	0.827	0.841	0.844	0.858	0.870	0.875
BDA (2 to 1)	0.748	0.765	0.756	0.797	0.809	0.812	0.853	0.821	0.859	0.865
GMM (1)	0.734	0.735	0.746	0.756	0.795	0.821	0.833	0.851	0.862	**0.885**
IWLSPC (3 to 2)	0.769	0.787	0.794	0.810	0.818	0.834	**0.857**	**0.868**	**0.873**	0.876
BDA (3 to 2)	0.730	0.778	0.790	0.802	0.804	0.841	0.838	0.862	0.838	0.847
IWLSPC (1 to 2)	**0.796**	**0.816**	**0.819**	**0.827**	**0.842**	0.837	0.835	0.850	0.868	**0.898**
BDA (1 to 2)	0.749	0.739	0.760	0.792	0.821	**0.853**	0.856	0.859	0.870	0.864
GMM (2)	0.720	0.728	0.751	0.763	0.790	0.786	0.804	0.843	0.865	0.876
IWLSPC (1 to 3)	**0.765**	0.771	0.783	0.788	0.806	0.818	0.829	0.853	0.862	**0.870**
BDA (1 to 3)	0.754	**0.794**	**0.824**	**0.844**	**0.865**	0.850	**0.862**	**0.885**	**0.870**	0.868
IWLSPC (2 to 3)	0.723	0.759	0.771	0.783	0.801	0.816	0.824	0.835	0.853	0.858
BDA (2 to 3)	0.745	0.789	0.797	0.835	0.827	**0.859**	0.841	0.851	0.855	0.861
GMM (3)	0.701	0.731	0.748	0.776	0.792	0.816	0.827	0.833	0.849	0.852

Bold values indicate the best performance in the comparative study.

TABLE IV
EXPERIMENTAL RESULTS FOR NATURALISTIC DRIVING DATA (EXPERIMENTAL II)

Number of Target Data	120 samples (10%)	240 samples (20%)	360 samples (30%)	480 samples (40%)	600 samples (50%)	720 samples (60%)	840 samples (70%)	960 samples (80%)	1080 samples (90%)	1200 samples (100%)
IWLSPC (3 to 1)	0.794	0.790	0.801	0.799	0.818	0.839	0.845	0.853	0.858	0.862
BDA (3 to 1)	**0.801**	0.783	0.790	**0.842**	**0.850**	0.837	0.835	0.862	0.870	**0.881**
IWLSPC (2 to 1)	0.788	**0.806**	**0.812**	0.827	0.835	**0.847**	**0.858**	**0.874**	**0.886**	0.879
BDA (2 to 1)	0.768	0.778	0.780	0.801	0.818	0.832	0.835	0.847	0.850	0.859
GMM (1)	0.663	0.689	0.707	0.724	0.765	0.794	0.835	0.845	0.847	0.853
IWLSPC (3 to 2)	0.771	0.800	0.806	0.830	0.829	0.850	0.847	0.857	0.855	0.872
BDA (3 to 2)	0.736	0.733	0.759	0.757	0.768	0.770	0.797	0.803	0.815	0.838
IWLSPC (1 to 2)	**0.815**	**0.833**	**0.838**	**0.849**	**0.867**	**0.856**	**0.862**	**0.874**	0.885	**0.890**
BDA (1 to 2)	0.727	0.733	0.738	0.771	0.765	0.803	0.783	0.818	0.829	0.844
GMM (2)	0.685	0.722	0.726	0.739	0.784	0.823	0.848	0.869	**0.892**	0.876
IWLSPC (1 to 3)	0.783	**0.812**	0.823	**0.829**	0.833	0.840	0.853	0.850	0.859	0.862
BDA (1 to 3)	0.736	0.754	0.750	0.780	0.785	0.827	0.832	0.850	**0.875**	0.881
IWLSPC (2 to 3)	**0.798**	0.806	0.818	**0.829**	0.835	0.847	0.867	0.866	0.872	0.875
BDA (2 to 3)	0.762	0.794	**0.827**	0.810	**0.867**	**0.886**	**0.890**	**0.873**	0.867	**0.892**
GMM (3)	0.660	0.713	0.742	0.771	0.810	0.830	0.866	0.860	0.873	0.870

Bold values indicate the best performance in the comparative study.

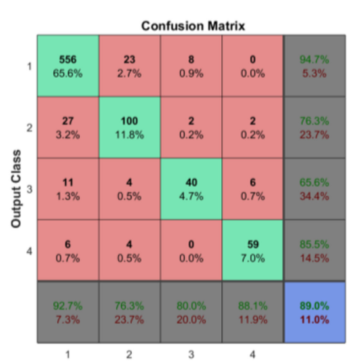


Fig. 9. Confusion matrix of recognized results based on IWLSPC.

0.6 to 0.2, and accuracies of recognition are 90.7%, 75.3%, 80.4%, and 84.1%. The comparative result for parameter tuning indicates that the optimal parameters selected by IWCV can represent the best performance of the trained model. Details of the comprehensive comparison are shown in Table IV.

In the simulated environment, we can simply ask the driver in the host vehicle to drive by requirements according to the information of the leading vehicle (only two vehicles in the environment: host and leading vehicles). However, for the naturalistic driving scenario, real road conditions are more complicated than those in the simulation. The driver's behavior is a combination of all perception information. And the impact factor used for simulation driving data can only represent the main influence. For this reason, comparing results in Figs. 6 and 7, the accuracies of naturalistic driving data are relatively lower than those in simulated driving data for each method in most conditions (with different training data).

G. Experiment III: Transfer Between Naturalistic Driving Data (Urban Scenarios)

In experiment I, the driving data from simulated environment is collected for verification. Considering that the scenario in the simulator is simpler and more controllable compared to real-world scenarios, the UAH-DriveSet is used to extract the

TABLE V
EXPERIMENTAL RESULTS FOR URBAN CAR-FOLLOWING DRIVING DATA (EXPERIMENTAL III)

Number of training data in the target domain	240 samples (20%)	600 samples (50%)	840 samples (70%)	1200 samples (100%)
IWLSPC (2 to 1)	**0.748**	**0.786**	**0.865**	**0.870**
BDA (2 to 1)	0.721	0.748	0.792	0.838
GMM (1)	0.654	0.710	0.759	0.827
IWLSPC (1 to 2)	**0.762**	**0.790**	**0.810**	**0.844**
BDA (1 to 2)	0.719	0.748	0.786	0.838
GMM (2)	0.646	0.716	0.707	0.815

Bold values indicate the best performance in the comparative study.

naturalistic driving data in experiment II. In order to fully support and verify the proposed DSBR framework, naturalistic driving data collected in the urban scenario are applied in experiment III. The platform for data collection is the same as the intelligent vehicle presented in [22].

In experiment III, two drivers with different driving experiences participate in the on-road data collection. The setting and procedure of data preprocessing are the same as experiments I and II. Experimental results are shown in Table V. With the increase of samples in the target domain, the recognized accuracy of all three methods increases. For the TL from driver 1 to driver 2, the performance of IWLSPC increases from 0.748 to 0.870, which is better than BDA and GMM. Also, the TL from driver 2 to driver 1 has the same trend. It indicates that the proposed DSBR framework can successfully model the transferable driver behavior in the urban scenario. Meanwhile, the time cost in the training and testing process is recorded to reflect the real-time performance. In experiment III, with 1200 samples in source and target domains, the training time of DSBR is 11.102 s, whereas the testing time for the recognition of BIL is about 8500 Observation/s. The result of the time cost indicates that the proposed method is acceptable for the real-time application.

V. CONCLUSION

In this article, a novel instance-level TL-based framework to recognize the driver's BIL in the car-following scenario was proposed and developed. The proposed framework combined an unsupervised method, GMM, and IWLSPC. First, GMM was applied to label BIL according to the distribution of driving data. Then, the TL-based method IWLSPC was designed to transfer the knowledge from the driver with sufficient driving data (source domain) and identify BIL for the new driver with insufficient driving data (target domain). In order to verify the proposed DSBR framework, naturalistic driving data collected from UAH-dataset were used. Experimental results indicate that the proposed DSBR framework has better performance than the method without TL when training data are not sufficient. It shows the advantages of the TL-based driver-specific behavior model. The proposed DSBR framework can model a new driver with insufficient driving data, which is difficult to realize by GMM and other conventional methods without TL. Meanwhile, reducing the reliance on the number of driving data has great potential in saving cost for driving data collection. This work pays more attention to the longitudinal driver behavior. In our future work, the lateral driver behavior will be considered.

Kumar *et al.* [24] and Rajaram and Subramanian [25] proposed to model an electropneumatic braking system by approximating the first-order linear system based on the mathematic method, which is used to predict the change of pressure in the braking chamber. The deducing process of time delay between the driver's braking actions and the change of deceleration is utilized to illustrate the rationality of letting deceleration represent braking behaviors. Experimental results indicated that the steady pressure in the braking chamber and the voltage supplied for the braking system (actuator) has a linear relationship

$$p_{\text{bss}} = 90000V_{\text{reg}} + p_{\text{atm}} \tag{32}$$

where V_{reg} is the voltage supplied to the electropneumatic regulator system and p_{atm} is the atmospheric pressure. The transient pressure's governing equation can be assumed as follows:

$$a_1\dot{\tilde{p}}_b(t) + \tilde{p}_b(t) = a_2u(t-\tau) \tag{33}$$

where $a_1$and a_2 can be obtained by experiments, $\tilde{p}_b(t)$ is the measuring pressure in the braking chamber, and $u(t-\tau)$ is the voltage supplied for the braking system. Finally, the open-loop transfer function is obtained as follows:

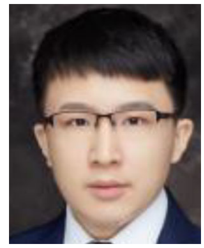
$$G(s) = \frac{\tilde{p}_b(s)}{U(s)} = \frac{a_2}{1+a_1s}e^{(-\tau s)} \tag{34}$$

where $e^{(-\tau s)}$ describes the time delay, which can be estimated and calculated by experiments. According to [24] and [25], the time delay for a step voltage is 30 ms.

REFERENCES

[1] P. Manfred and J. Edelmann, "Driver models in automobile dynamics application," *Veh. Syst. Dyn.*, vol. 45, no. 7/8, pp. 699–741, 2007.
[2] Y. Xing, C. Lv, D. Cao, and C. Lu, "Energy oriented driving behavior analysis and personalized prediction of vehicle states with joint time series modeling," *Appl. Energy*, vol. 261, 2020, Art. no. 114471.
[3] C. Lv, J. Zhang, and Y. Li, "Extended-Kalman-filter-based regenerative and friction blended braking control for electric vehicle equipped with axle motor considering damping and elastic properties of electric powertrain," *Veh. Syst. Dyn.*, vol. 52, no. 11, pp. 1372–1388, 2014.
[4] F. Bella and M. Silvestri, "Driver's braking behavior approaching pedestrian crossings: A parametric duration model of the speed reduction times," *J. Adv. Transp.*, vol. 50, no. 4, pp. 630–646, 2016.
[5] M. Da Lio, A. Mazzalai, K. Gurney, and A. Saroldi, "Biologically guided driver modeling: The stop behavior of human car drivers," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2454–2469, Aug. 2018.
[6] W. Li, H. Du, and W. Li, "Driver intention based coordinate control of regenerative and plugging braking for electric vehicles with in-wheel PMSMs," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1300–1311, 2018.
[7] W. Wang, J. Xi, and D. Zhao, "Learning and inferring a driver's braking action in car-following scenarios," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 3887–3899, May 2018.
[8] T. Teng, L. Bi, and Y. Liu, "EEG-based detection of driver emergency braking intention for brain-controlled vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1766–1773, Jun. 2018.
[9] I. H. Kim, J. W. Kim, S. Haufe, and S. W. Lee, "Detection of braking intention in diverse situations during simulated driving based on EEG feature combination," *J. Neural Eng.*, vol. 12, no. 1, 2015, Art. no. 016001.
[10] C. Lv *et al.*, "Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5718–5729, Jul. 2018.
[11] C. M. Martinez, M. Heuke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advance driver assistance: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, Mar. 2018.

[12] V. Butakov and P. Ioannou, "Personalized driver/vehicle lane change models for ADAS," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4422–4431, Oct. 2015.
[13] L. Li *et al.*, "Parallel testing of vehicle intelligence via virtual-real interaction," *Sci. Robot.*, vol. 4, no. 28, 2019, Art. no. eaaw4106.
[14] M. Taieb-Maimon and D. Shinar, "Minimum and comfortable driving headways: Reality versus perception," *Hum. Factors*, vol. 43, no. 1, pp. 159–172, 2001.
[15] L. Pariota, G. N. Bifulco, and M. Brackstone, "A linear dynamic model for driving behavior in car following," *Transp. Sci.*, vol. 50, no. 3, pp. 1032–1042, 2016.
[16] E. R. Boer, "Car following from the driver's perspective," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 2, no. 4, pp. 201–206, 1999.
[17] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Virtual-to-real knowledge transfer for driving behavior recognition: Framework and a case study," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6391–6402, Jul. 2019.
[18] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Transfer learning for driver model adaptation in lane-changing scenarios using manifold alignment," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3281–3293, Aug. 2020.
[19] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, 2009.
[20] 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.
[21] H. Hachiya, M. Sugiyama, and N. Ueda, "Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition," *Neurocomputing*, vol. 80, pp. 93–101, 2012.
[22] Z. Li, J. Gong, C. Lu, and J. Xi, "Importance weighted Gaussian process regression for transferable driver behaviour learning in the lane change scenario," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12497–12509, Nov. 2020.
[23] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2016, pp. 387–392.
[24] Y. S. R. R. Kumar, D. B. Sonawane, and S. C. Subramanian, "Application of PID control to an electro-pneumatic brake system," *Int. J. Adv. Eng. Sci. Appl. Math.*, vol. 4, no. 4, pp. 260–268, 2012.
[25] V. Rajaram and S. C. Subramanian, "A model based collision avoidance algorithm for heavy commercial vehicles," in *Proc. Amer. Control Conf.*, 2014, pp. 3213–3218.
[26] A. Lopez, R. Sheroni, S. Chien, L. Li, Q. Yi, and Y. Chen, "Analysis of the braking behaviour in pedestrian automatic emergency braking," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2015, pp. 1117–1122.



Zirui Li received the B.S. degree from the Beijing Institute of Technology, Beijing, China, in 2019, where he is currently working toward the Ph.D. degree in mechanical engineering.

His research interests include intelligent vehicles, driver behavior modeling, and transfer learning.

Jianwei Gong (Member, IEEE) received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2002.

He is currently a Professor and the Director with the Intelligent Vehicle Research Centre, School of Mechanical Engineering, Beijing Institute of Technology. His research interests include intelligent vehicle decision-making and path/motion planning and control.

Chao Lu (Member, IEEE) received the Ph.D. degree in transport studies from the University of Leeds, Leeds, U.K., in 2015.

He is currently an Associate Professor with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China. His research interests include driver behavior modeling, reinforcement learning, and transfer learning and its applications.

Jinghang Li received the M.S. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2021.

His research interests include intelligent vehicles, machine learning, and manifold learning.