Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 4752651, 13 pages
https://doi.org/10.1155/2020/4752651

Research Article

Learning the Car-following Behavior of Drivers Using Maximum Entropy Deep Inverse Reinforcement Learning

Yang Zhou,[1,2] **Rui Fu**,[1] **and Chang Wang**[1]

[1]*School of Automobile, Chang'an University, Middle Section of Nan Erhuan Road, Xi'an 710064, China*
[2]*School of Vehicle Engineering, Xi'an Aeronautical University, No. 259, Xi Erhuan Road, Xi'an 710077, China*

Correspondence should be addressed to Rui Fu; furui@chd.edu.cn

Received 29 January 2020; Revised 6 September 2020; Accepted 23 October 2020; Published 21 November 2020

Academic Editor: Lelitha Vanajakshi

The present study proposes a framework for learning the car-following behavior of drivers based on maximum entropy deep inverse reinforcement learning. The proposed framework enables learning the reward function, which is represented by a fully connected neural network, from driving data, including the speed of the driver's vehicle, the distance to the leading vehicle, and the relative speed. Data from two field tests with 42 drivers are used. After clustering the participants into aggressive and conservative groups, the car-following data were used to train the proposed model, a fully connected neural network model, and a recurrent neural network model. Adopting the fivefold cross-validation method, the proposed model was proved to have the lowest root mean squared percentage error and modified Hausdorff distance among the different models, exhibiting superior ability for reproducing drivers' car-following behaviors. Moreover, the proposed model captured the characteristics of different driving styles during car-following scenarios. The learned rewards and strategies were consistent with the demonstrations of the two groups. Inverse reinforcement learning can serve as a new tool to explain and model driving behavior, providing references for the development of human-like autonomous driving models.

1. Introduction

Recent studies have suggested that the development of autonomous driving may benefit from imitating human drivers [1–3]. There are two reasons: First, the comfort of autonomous vehicles (AVs) may be improved if the driving styles match the preferences of the passengers. Second, the transition period during which AVs will share the road with human-driven cars is expected to last for decades. Road safety may be enhanced if AVs are designed to understand how human drivers will react in different situations.

Car-following is one of the most common situations encountered by drivers. The modeling of car-following behavior has been a common research focus in the fields of traffic simulation [4], advanced driver-assistance system (ADAS) design [5], and connected driving and autonomous driving [6–9]. Various car-following models have been proposed since 1953 [10]. In general, there are two major approaches. The classical methods use several parameters to characterize the car-following behavior of drivers [11, 12]. With the rapid development of data science, data-driven methods with a focus on learning the behavior of drivers based on field data [13, 14] have emerged. For both approaches, data-driven car-following models were found to provide the highest accuracy and best generalization ability for replicating the drivers' trajectories.

Among data-driven methods, supervised learning and expressive models, such as neural networks (NNs), have been commonly used to learn the relationships between states and drivers' controls [15–17]. These modeling techniques are often referred to as behavior cloning (BC). Even though BC approaches have been successfully applied, they are prone to cascading errors [18], which is a well-known problem in the sequential decision-making literature. The reason is that inaccuracies occur in model predictions when there are insufficient data for training the model. Small inaccuracies accumulate during the simulation, which

eventually leads the model to states not included in the training data and brings about even poorer predictions.

Inverse reinforcement learning (IRL) was introduced to overcome these drawbacks. IRL, which was proposed by Ng and Russell [19], is the inverse problem of reinforcement learning (RL). Although RL has been applied with great success in recent years, such as in the well-known program AlphaGo [20], the use of RL in other domains remains limited because it is challenging to determine the reward, which is the core component in RL. Manual tweaking of the reward functions can be tedious, and inappropriate reward assignments may lead to unexpected behaviors [21]. IRL, however, provides a framework to learn the rewards automatically. The advantages of IRL are twofold: the learned rewards can be used to improve the interpretability of the models, and the goals of the tasks can be understood, which may prevent cascading errors [22]. Therefore, the present study proposes a car-following model based on IRL. In contrast to a recent work, which applied IRL to model car-following using linear reward representation [23], in this study, a nonlinear function, that is, NN, is used to approximate the reward function as the preferences of human drivers may be highly nonlinear. The proposed model is trained and tested using data under actual driving conditions, and the performance is compared with that of other car-following models.

The rest of the paper is organized as follows: Section 2 briefly reviews the literature on car-following modeling, RL, and IRL. Section 3 presents the input feature vectors of the reward network in the IRL and the proposed algorithm. Section 4 describes the experiments and data used in this study. Section 5 elaborates on the training process of the proposed model and presents the investigated car-following models. Section 6 presents the comparison of the performance for different methods and the characteristics of the trained models using data from drivers with different driving styles. The final section presents the discussion and conclusion.

2. Background

The car-following process is essentially a sequential decision-making problem where drivers continually adjust the longitudinal control a based on the states s they encounter, which include the speed of the driver's car, the spacing between the driver's car and the leading car, and the relative speed between the two vehicles. Car-following models are designed to model the policy $\pi(a|s)$ of drivers.

2.1. Classical Car-following Models. The early General Motors models proposed by Chandler [24] modeled the drivers' longitudinal controls to minimize the relative speed because this is one of the primary objectives of car-following. These models exhibited poor performance in predicting the distance between cars. Later models addressed this problem by considering another objective of car-following, that is, maintaining the desired distance; these models included the Gipps model [25] and the intelligent driver model (IDM) [12].

2.2. Behavior Cloning Car-following Models. As the access to high-fidelity driving data has become increasingly available, data-driven models such as NN have been used to model car-following behavior. NN have been demonstrated to exhibit excellent performance for estimating nonlinear and complex relationships. In 2003, Jia et al. [16] proposed an NN-based car-following model with two hidden layers and the inputs speed, relative speed, spacing, and desired speed. Chong et al. [15] simplified the architecture proposed by Jia to one hidden layer and obtained similar results. Instead of using as input only a single time step of relevant information, such as in the conventional NN-based models, Zhou et al. [17] proposed a recurrent neural network- (RNN-) based model that used a sequence of past driving information as input. The RNN approach was better adapted to changes in traffic conditions than the NN approaches. The present study also uses the RNN-based model to compare its performance with that of the proposed method.

2.3. Reinforcement Learning. In RL, a sequential decision-making problem is modeled as a Markov-decision process (MDP), which is defined as a tuple $M = \{S, A, T, r, \gamma\}$. S and A denote the state and action space, respectively, and T denotes the transition matrix, which is defined in equation (1). r and γ denote the reward function and the discount factor, respectively.

$$\begin{aligned} v(t+1) &= v(t) + a(t) * \Delta t, \\ \Delta v(t+1) &= v_{\text{lead}}(t+1) - v(t+1), \\ h(t+1) &= h(t) + \frac{\Delta v(t) + \Delta v(t+1)}{2} * \Delta t, \end{aligned} \tag{1}$$

where $v(t)$, $\Delta v(t)$, and $h(t)$ denote the speed of the ego vehicle, the relative speed from the lead vehicle, and the spacing between the ego and the leader at time step t, respectively. Δt is the simulation time interval, which is 0.1 s in this study, and v_{lead} denotes the speed of the lead vehicle, which was obtained from the collected data.

RL assumes that drivers follow a policy that maximizes long-term rewards. Once the rewards are known, the policy can be determined using algorithms such as Q-learning [26]. In recent years, RL has been applied by researchers to solve real-world problems such as the balance control of a robot and the energy management of hybrid electric vehicles [27–29].

2.4. Inverse Reinforcement Learning. In IRL, the reward of a state can be represented by a linear combination of the relevant features (equation (2)). The goal of IRL is to determine the weights θ from expert demonstrations.

$$r(s) = \theta^{\mathrm{T}} f(s). \tag{2}$$

Abbeel and Ng [30] proposed a feature matching strategy to solve the problem (equation (3)). As long as the feature expectation of the simulated trajectories equals the features calculated from the expert data, the learned behavior has the same performance as the demonstrator. However, it was found that many different policies can be obtained when the feature matching conditions were satisfied. The ambiguity problem related to the correct reward and policy remains unsolved.

$$\begin{aligned} \sum_{\tau} p(\tau) f(\tau) &= \tilde{f}, \\ p(\tau) &= p(s_1, a_1, \ldots s_T, a_T), \\ &= p(s_1) \prod_{t=1}^{\mathrm{T}} \pi(a_t | t s_t) T(s_{t+1} | s_t, a_t). \end{aligned} \tag{3}$$

The maximum entropy IRL (Max-Ent IRL) proposed by Ziebart [31] addressed the ambiguity problem by incorporating the principle of maximum entropy into the IRL. In the Max-Ent IRL framework, the probability of a trajectory is proportional to the sum of the exponential rewards accumulated in the trajectory (equation (4)). This form of distribution can guarantee no additional preferences other than the feature matching requirement. When the probability of a trajectory is known, the weights of the reward can be determined by maximizing the log-likelihood of the expert data using the following objective function (equation (5)):

$$p(\tau) \propto e^{\sum_{s_i \in \tau} \theta^{\mathrm{T}} f(s_i)}, \tag{4}$$

$$\theta^* = \mathrm{argmax}_{\theta} \log(p(\tau_D)). \tag{5}$$

2.5. Maximum Entropy Deep Inverse Reinforcement Learning. Since the linear representation of the rewards might limit the accuracy of reward approximation, Wulfmeier [32] extended the method to nonlinear models using deep NNs. Deep architectures have been shown to capture the nonlinear reward structure in several benchmark tasks with high accuracy. The present study uses the approach of deep architectures to represent the rewards of drivers in car-following. The fully connected NNs used in this study map the input features in the car-following model to estimate the rewards, as shown in Figure 1.

It can be derived that the gradient of the Max-Ent deep IRL (DIRL) is as follows:

$$\mathrm{grad} = \left(\mu_D - E_{\mu}\right) \frac{d}{d\theta} g(f, \theta), \tag{6}$$

where μ_D and E_{μ} refer to the state visitation frequencies calculated from the expert demonstrations and expected state visitation frequencies obtained from the learned policy and $g(f, \theta)$ refers to the network architectures. Once the gradient is calculated, the parameters of the NN are updated using backpropagation [33].

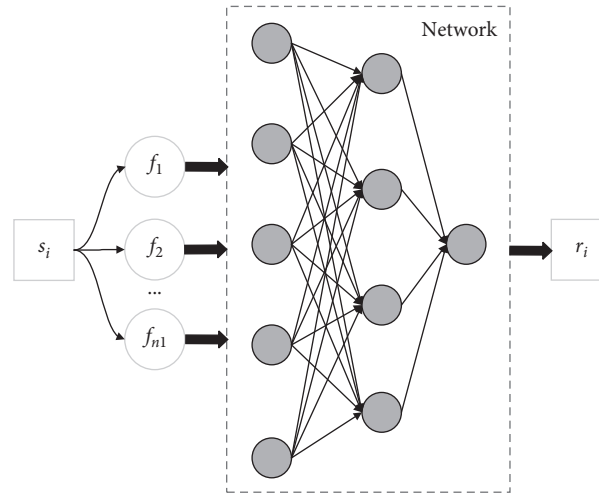


Figure 1: The neural network used to approximate the rewards.

3. The Proposed Car-following Model

In this section, the details of the proposed model (DIRL) are explained, including the design of the input features for the reward network and the full algorithm. The DIRL model uses as input the driver data on car-following trajectories, consisting of speed during car-following, spacing to the leading car, and relative speed. After training, the DIRL model outputs the policy and the rewards of drivers. A discrete state and action space were defined in the present study. According to the rules for determining car-following events that will be described in Section 4.2 and the distribution of the empirical data used in this study, the spacing h is limited to the range from 0 to 120 m with an interval of 0.5 m, the speed v is limited to the range from 0 to 33 m/s with an interval of 0.5 m/s, and the relative speed Δv is limited to the range from −5 to 5 m/s with an interval of 0.5 m/s. The action a is limited to the range from -3 to 2 m/s^2 with an interval of 0.2 m/s^2.

3.1. Feature Selection for the Rewards in Car-following. As introduced in the last section, the input features of the network are determined first to create an NN and obtain the rewards in car-following. The rewards in RL encode the objectives or the purpose of the agent [26]. Therefore, the selected features should represent the objectives of drivers in the car-following task.

In the study of Gao [23], speed and spacing were chosen as features for representing the rewards. In [34], the reward function represented the speed discrepancies between the simulated trajectories and the test data. In contrast to these studies, we base the reward function on the following features.

3.1.1. Time-Headway. Time-headway (TH) has been widely used as an indicator for drivers to evaluate risk during car-following [35]; TH is defined as the time between two vehicles passing the same point on the road. It has been suggested that a driver's safety margin in car-following can

be characterized by the TH, which plays a role in the driver's decision-making [36]. Drivers may have different desired safety margins for the TH. For example, aggressive drivers may prefer a shorter TH than conservative drivers because they like to track vehicles at a closer distance. It has been suggested that one of drivers' objectives in car-following is to control TH to their expectations [37]. Therefore, TH is selected as an input of the reward network in this study.

3.1.2. Relative Speed. Research has shown that the drivers' speed control in car-following is proportional to the relative speed [38]. As mentioned earlier, an objective in car-following is to keep the relative speed close to zero [37]. In this study, we relax this objective so that drivers will keep the relative speed within an appropriate range because people's driving behavior is imperfect and is not always optimal.

Following the method presented in [23], these two features were mapped into high-dimensional space using the Gaussian radial kernel:

$$f_1(s) = \exp\left(-\frac{(s - s_i)}{\sigma^2}\right), \tag{7}$$

where $s_i = (\mathrm{TH}_i, \Delta V_i)$ denotes the kernel vectors, which represent the conjectural values of the preferred TH and relative speed, and σ is a parameter that controls the width of the kernel function. Specifically, TH_i has a range of 0.5 s to 3 s, with an interval of 0.5 s, and ΔV_i has a range of −4 m/s to 4 m/s, with an interval of 0.5 m/s in this study.

3.1.3. Maximum Speed. The maximum desired speed is commonly used in many classical car-following models [12, 16]. Drivers may have a preferred maximum speed, and they may not continue to follow the leader if their speed is already above this value. It is assumed that the objective of the driver is to keep the speed below the maximum speed as follows:

$$f_2(s) = \begin{cases} 1, & v \le v^i_{\max}, \\ 0, & v > v^i_{\max}, \end{cases} \tag{8}$$

where $v^i_{\max}$ denotes the conjectural acceptable maximum speed. $v^i_{\max}$ is in the range of 90 km/h to 120 km/h, with an interval of 5 km/h. The reward function is represented by an NN that is parameterized by θ as follows:

$$r(s) = g(f_1 \cdot f_2, \theta). \tag{9}$$

3.2. The Full Algorithm. The proposed DIRL algorithm consists of three parts, which are marked in bold in Algorithm 1. In the first part, the reward $r^i(s)$ is determined by the parameters of the NN to calculate the policy $\pi^i(a|s)$. Value iteration with a softmax function is used to solve the policy based on the reward. The result of the softmax version of value iteration is a stochastic policy in which the probabilities of every predefined action are listed in a tabular form. $V(s)$ and $Q(s, a)$ in this part denote the expected long-term return of states and state-action pairs.

In the second part, the policy $\pi^i(a|s)$ is applied to estimate the expected state visitation frequencies $\mu^i(s)$. The original version for estimating $\mu^i(s)$, as reported in [31], is not suitable in car-following tasks because the speed of the lead vehicle is always changing. Simply applying policy propagation [32] for every trajectory in the data can be time-consuming. Therefore, in this study, we perform sampling by running the policy in the simulation of drivers' car-following trajectories for N_2 times to approximate $\mu^i(s)$. During the simulation, the action at every time step was randomly sampled from the policy based on the probability of every action.

In the third part, the gradients are calculated by subtracting the estimated $\mu^i(s)$ from the state visitation frequencies μ_D obtained from the data. Subsequently, the parameters of the NN are updated by backpropagation. These steps are repeated several times until convergence. The training of the algorithm can be stopped when the rewards accumulated in the trajectories stop increasing.

4. Experiments

4.1. Data Description. Data from two field tests that were conducted in Huzhou city in Zhejiang province and Xi'an city in Shaanxi province were used in this study. Forty-two drivers participated in the test. Their driving experience ranged from 2 to 30 years with the average being 15.2 years. During the test, the participants were only informed of the starting location and destination, and they were asked to follow their normal driving styles. The test data were collected by a Volkswagen Touran equipped with instruments and sensors, as illustrated in Figure 2. The test route consisted of diverse driving scenarios such as urban roads and highways, as shown in Figure 3. The other details of the field tests are described in [39, 40].

4.2. Extraction of Car-following Events and Data Filtering. We applied the rules described in [41] to extract the car-following events from the obtained data. (1) We ensured that the test vehicle was following the same lead car; (2) the distance to the lead car was less than 120 m to eliminate free-flow traffic conditions; (3) we ensured that the follower and the leader were on the same lane; (4) the duration of car-following events was longer than 15 s.

The extracted events were then manually reviewed by checking the videos recorded by the front camera on the equipment vehicle to guarantee good data quality. Eventually, nearly one thousand car-following events were extracted. A moving average filter was applied (1 s) to remove noise from the extracted car-following data.

4.3. Driving Style Clustering. The participants displayed diverse driving styles, which were evident in the driving data. The k-means algorithm was used to cluster the drivers into different driving styles. Previous studies have adopted kinematic features such as spacing, speed, and relative speed or time-based features such as TH and TTC for driving style clustering [34, 39]. In this study, multiple combinations of

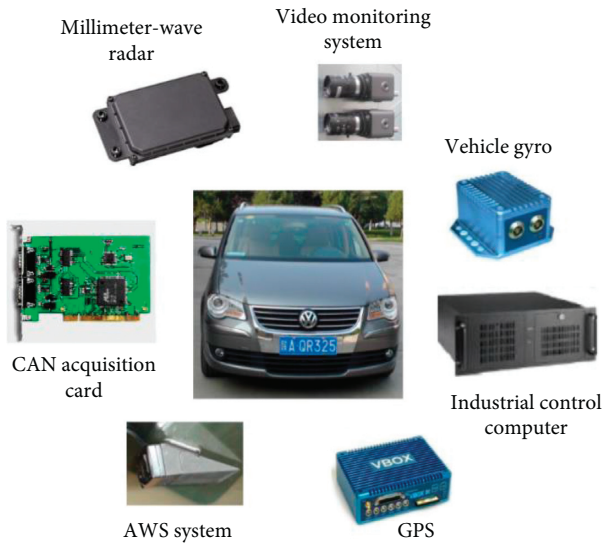


Figure 2: The vehicle and equipment used in the experiment.

the mentioned features were tested as inputs for the k-means algorithm, and the quality of the clustering results was then evaluated by the silhouette coefficient where a larger silhouette coefficient indicates a better result. Finally, the mean value of TH and TH when braking was chosen because this combination achieved the highest value of the silhouette coefficient [42]. The number of the clusters was also determined to be two based on the results of the silhouette coefficient. Figures 4 and 5 present the boxplot of the mean TH and mean TH when braking for the conservative group that consisted of 16 drivers and the aggressive group that consisted of 26 drivers, respectively. The aggressive group had significantly higher mean TH ($t = 6.748$, $p < 0.001$) and mean TH when braking ($t = 7.655$, $p < 0.001$) than the conservative group.

The descriptive statistics (Table 1) of the two groups confirmed the clustering results. The aggressive drivers had shorter mean spacing and higher mean speed and mean acceleration than the conservative drivers.

5. Model Training and Evaluation

5.1. Evaluation Metrics. Two metrics, the root mean square percentage error (RMSPE) (equation (10)) and the modified Hausdorff distance (MHD), were used to evaluate the accuracy of the car-following models for reproducing drivers' car-following trajectories. As suggested by Punzo and Montanino [43], the cumulative sum of the errors is an appropriate option to evaluate the performance of car-following models.

$$\text{RMSPE(speed)} = \sqrt{\frac{\sum_t \left[v_n^{\text{obs}}(t) - v_n^{\text{simu}}(t)\right]^2}{\sum_t \left[v_n^{obs}(t)\right]^2}},$$
$$\text{RMSPE(spacing)} = \sqrt{\frac{\sum_t \left[h_n^{\text{obs}}(t) - h_n^{\text{simu}}(t)\right]^2}{\sum_t \left[h_n^{\text{obs}}(t)\right]^2}}, \quad (10)$$

where RMSPE(speed) denotes the RMSPE of speed, RMSPE(spacing) denotes the RMSPE of spacing, $v_n^{\text{obs}}(t)$, $h_n^{\text{obs}}(t)$ are the speed and spacing at time t in the observed nth trajectory, and $v_n^{\text{simu}}(t)$, $h_n^{\text{simu}}(t)$ are the simulated speed and spacing at time t for the nth trajectory.

The MHD is an extension of the Hausdorff distance which represents the distance between two sets of points $C = \{c_1, c_2, \ldots, c_{N_c}\}$ and $B = \{b_1, b_2, \ldots, b_{N_b}\}$, as defined in equation (11). The median of the MHD (MHD_{50}) had been used to evaluate the similarity of simulated and actual trajectories in modeling defensive driving strategies [44] and urban route planning [45].

$$d(c, B) = \min_{b \varepsilon B} \|c - b\|,$$
$$d(C, B) = \frac{1}{N_c} \sum_{c \varepsilon C} d(c, B), \quad (11)$$
$$\text{MHD} = \max(d(C, B), d(B, C)).$$

Since the proposed DIRL model outputs a stochastic policy, the two metrics were calculated by averaging the results of 10 simulations for every trajectory in the data.

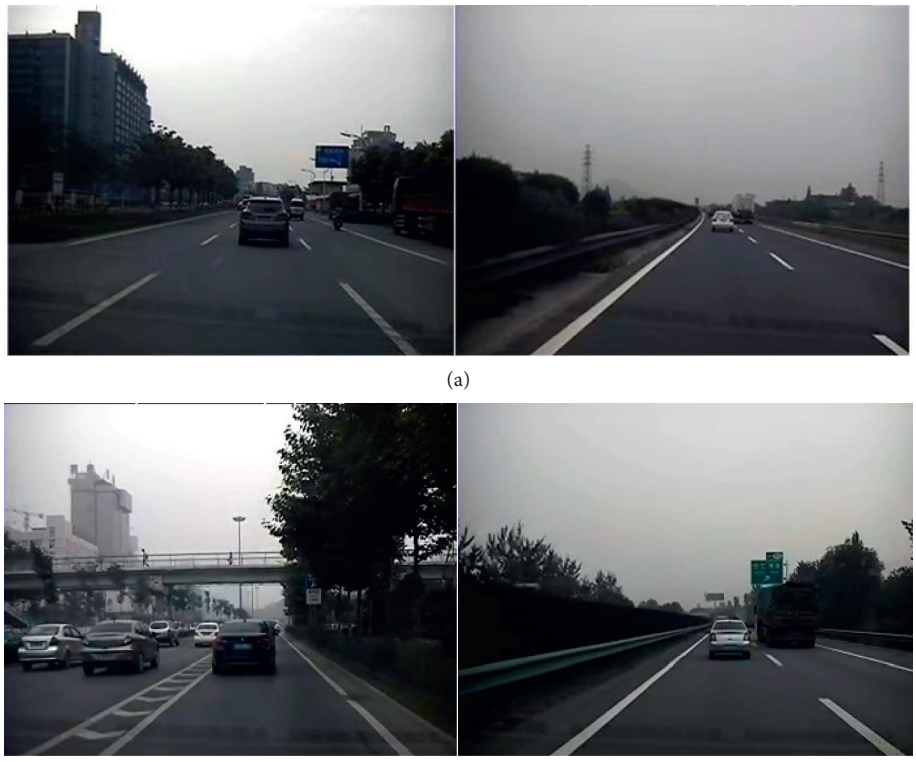
5.2. Model Training. The k-fold cross-validation method was applied to evaluate the performance of the car-following models. Specifically, the car-following datasets of the two groups of drivers were randomly divided into 5 groups with an equal number of trajectories. One group was taken as the test set and the remaining four groups were taken as the training set. The training and test experiments were repeated five times because every divided group had been used as the test set. Finally, the performance of the car-following models was evaluated by the average value of the two metrics.

The Adam optimizer [46] with learning rate decay was applied to train the DIRL model. The hyperparameters used for training are listed in Table 2. L2 regularization was used to prevent overfitting of the reward network.

Figures 6 and 7 present the change of RMSPE of spacing and the change of the cumulative normalized rewards per trajectory in one of the cross-validation experiments, respectively. After about 5 iterations, the RMSPE of spacing for the training set and test set start to converge. The rewards collected in the trajectory remain stable after about the same number of iterations.

5.3. The Investigated Models. The accuracy and generalization ability of the proposed model was compared with those of two other data-driven car-following models, that is, the NN-based model and the RNN-based model.

5.3.1. NN-Based Car-following Model. A fully connected neural network with one hidden layer was built following the study conducted by Chong et al. [15]. The hidden layer consisted of 60 neurons in this study. The NN-based model takes inputs of speed, spacing, and relative speed and outputs the acceleration for the current time step. The objective



(a)

(b)

Figure 3: Driving scenarios in (a) Huzhou city and (b) Xi'an city.

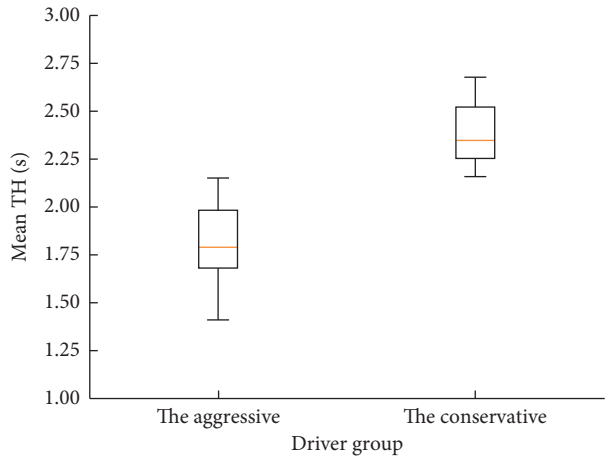


Figure 4: The boxplot of mean TH for the two groups of drivers.

of minimizing the empirical acceleration and the model's predictions was adopted to train the model (equation (12)).

$$L(w,b) = \left(a_n^{\text{simu}}(t) - a_n^{\text{obs}}(t)\right)^2, \tag{12}$$

where w, b denotes the weights and bias in the NN-based model, $a_n^{\text{simu}}(t)$ denotes the predicted acceleration at time step t for the nth trajectory, and $a_n^{\text{obs}}(t)$ denotes the empirical acceleration at time step t for the nth trajectory.

5.3.2. RNN-Based Car-following Model. The architecture of the RNN-based model built in this study is in line with the study conducted by Zhou et al. [17]. The number of hidden neurons in the RNN model was set to be 60. The RNN model takes inputs of a sequence of historical information that lasts for 1 s and outputs the acceleration for the current time step. The speed and spacing for the next time step were then estimated based on the state transition matrix described in equation (1). The training of the RNN model adopted the

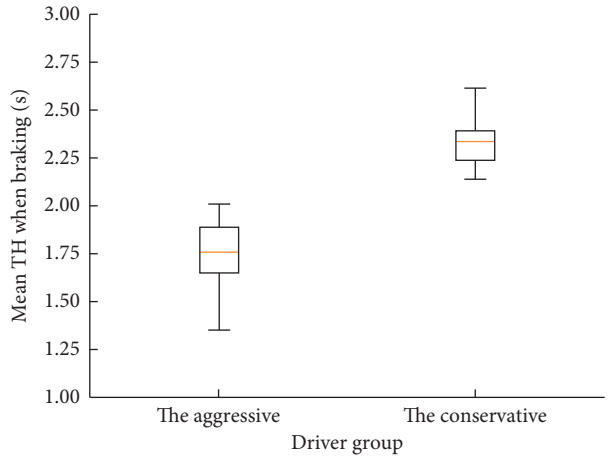


Figure 5: The boxplot of mean TH when braking for the two groups of drivers.

Input: f_1, f_2, T, σ, μ_D, γ
Randomly initialize the parameters of the neural network as θ^1
For $i = 1$ to N_1 do
  Determine the reward for every state by applying forward propagation in the neural network
  $r^i(s) = g(f_1 \cdot f_2, \theta^i)$
  Use the softmax version of value iteration to obtain the policy
    Initialize $V(s) = -\infty$
    Repeat until $\max(V(s) - V_t(s)) < \varepsilon$
      $V_t(s) = V(s)$
      $Q(s,a) = r^i(s) + \gamma * E_{T(s,a,s\prime)} V(s\prime)$
      $V(s) = \sigma * \log \int \exp(Q(s,a)/\sigma) da$
    $\pi^i(a|s) = \exp(Q(s,a) - V(s))$
  Estimate the expected state visitation frequencies $\mu^i(s)$ using the policy π^i
    For $j = 1$ to N_2 do
      Start from the initial state in every trajectory and run the policy π^i
      For every time step, sample one action from the distribution of π^i according to the probability of every action
      $a = \text{random\_sample}(p = \pi^i(a|s))$
      $s\prime = T(s, a, s\prime)$
      $\mu^i(s\prime) += 1$
    end for
    $\mu^i(s) = \mu^i(s)/N_2$
  Calculate the gradients of DIRL and the network and use backpropagation to update the parameters of the network
    $\text{grad}_r^i = \mu_D - \mu_s^i$
    $\text{grad}_\theta^i = \text{back\_propagation}(\text{grad}_r^i, \theta^i)$
    Update θ^i with the gradients grad_θ^i
end for

Algorithm 1: DIRL: Maximum entropy deep inverse reinforcement learning for car-following modeling.

Table 1: Descriptive statistics for different driving styles.

Type	Spacing (m)			Speed (m/s)			Relative speed (m/s)			Acceleration (m/s^2)		
	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
Aggressive	34.15	2.27	120.00	18.57	2.50	33.70	−0.42	−12.45	5.80	−0.01	−3.14	1.64
Conservative	42.79	3.91	119.90	17.47	1.49	34.84	−0.52	−8.85	5.02	−0.04	−1.80	1.40

Table 2: The hyperparameters used for training.

Hyperparameters	Value
Learning rate	0.0005
Learning rate decay	0.95
Reward discount (γ)	0.95
Temperature (σ)	0.7

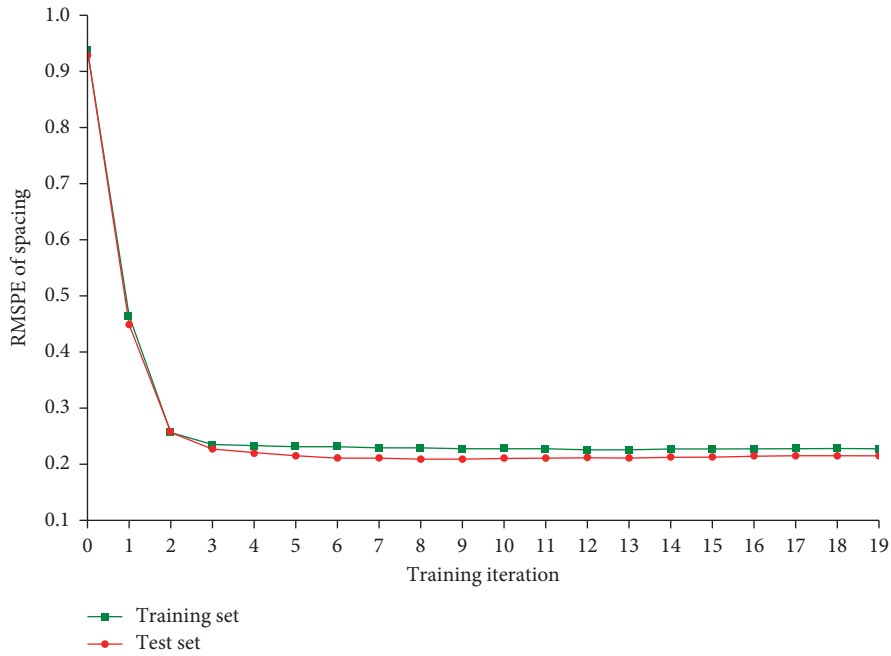


Figure 6: The change of RMSPE of spacing during training.

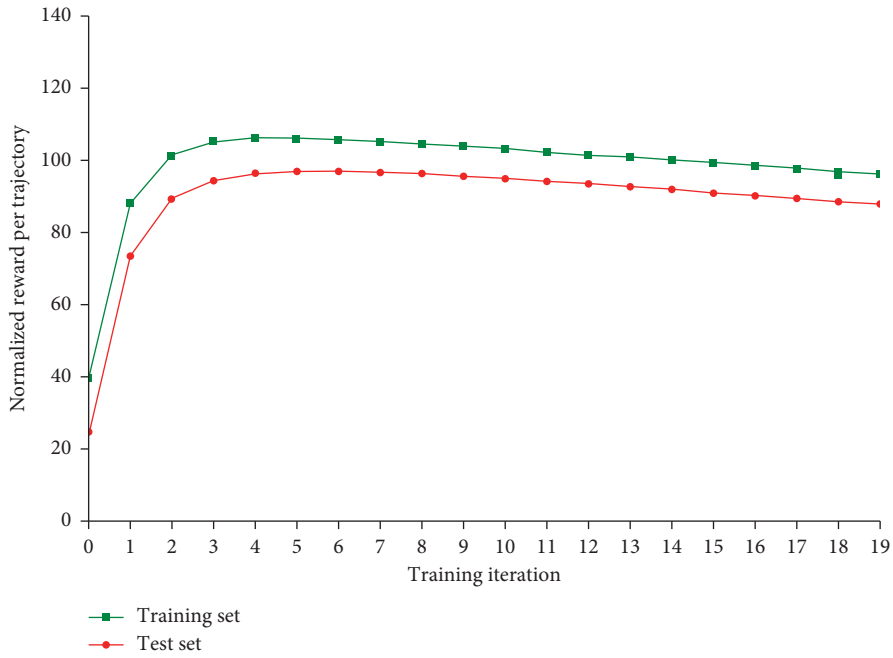


Figure 7: The change of the normalized reward accumulated in every trajectory during training.

Table 3: The average performance of the models on the training sets.

	RMSPE of spacing (%)		RMSPE of speed (%)		MHD_{50}	
	Aggressive	Conservative	Aggressive	Conservative	Aggressive	Conservative
NN	29.07	27.71	5.88	5.97	2.75	2.92
RNN	28.18	23.82	6.82	6.14	2.72	2.94
DIRL	22.83	23.48	6.57	7.08	2.68	2.85

Table 4: The average performance of the models on the test sets.

	RMSPE of spacing (%)		RMSPE of speed (%)		MHD_{50}	
	Aggressive	Conservative	Aggressive	Conservative	Aggressive	Conservative
NN	28.32	27.01	6.09	6.01	2.83	2.98
RNN	23.99	25.53	5.58	6.54	2.67	2.84
DIRL	21.58	22.15	6.14	7.46	2.64	2.77

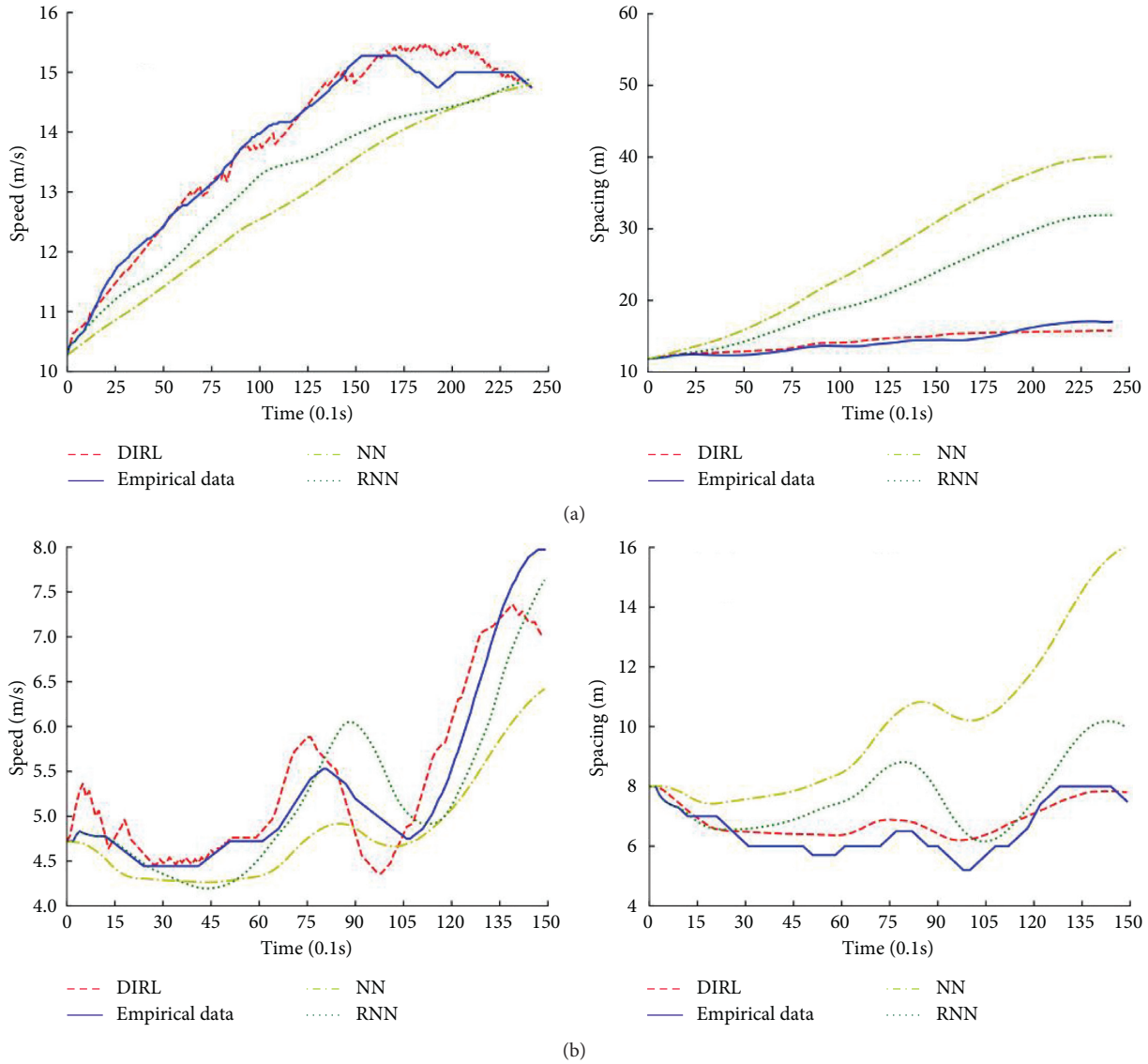


Figure 8: The simulation results of speed (left) and spacing (right) for two car-following periods (a) and (b) by different models.

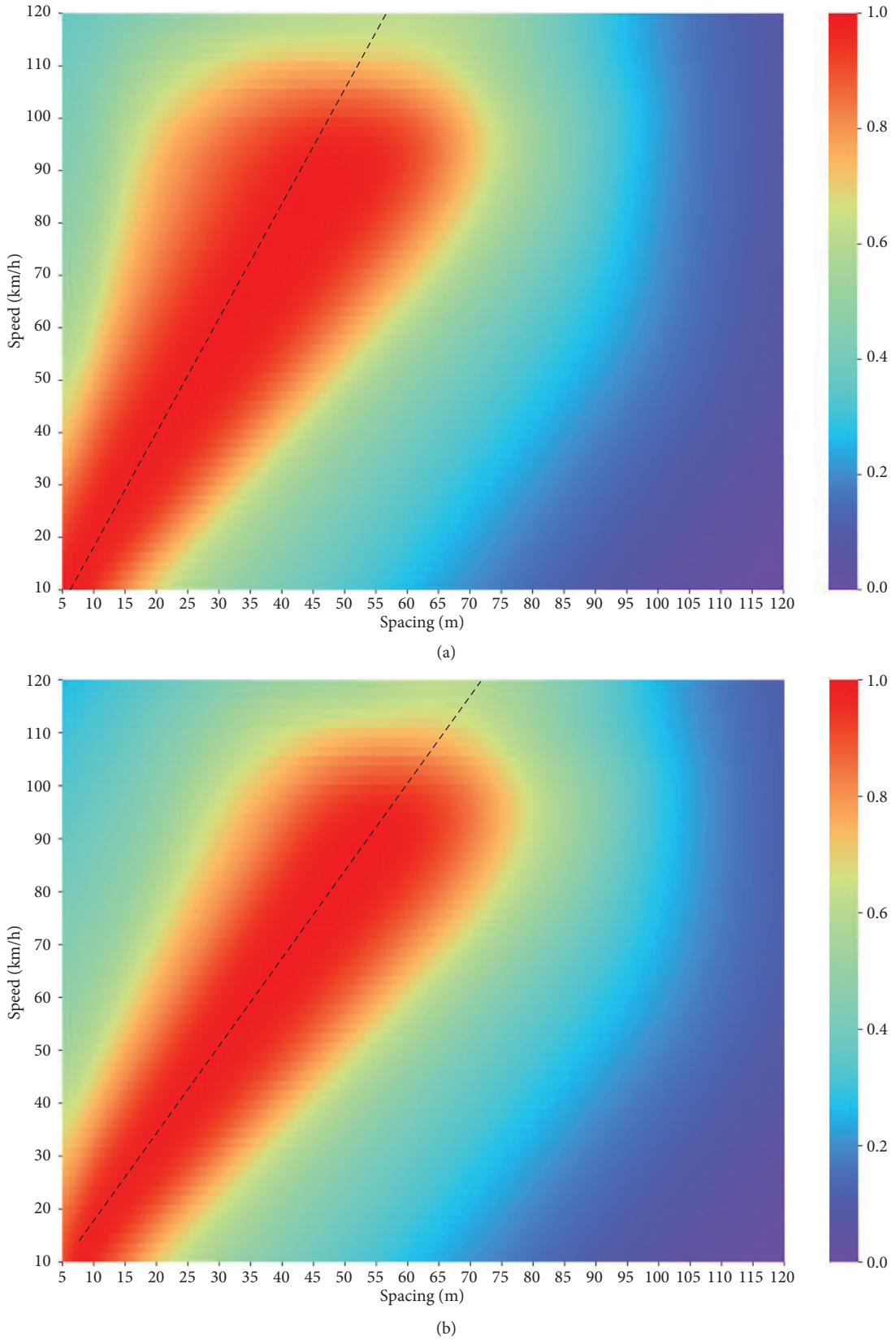


Figure 9: The comparison of the value (V) for (a) the aggressive drivers and (b) the conservative drivers. The colors indicate the value of every state in the state space, and the black-dashed line lies in the center of the high-value area, indicating the direction of the high-value area.

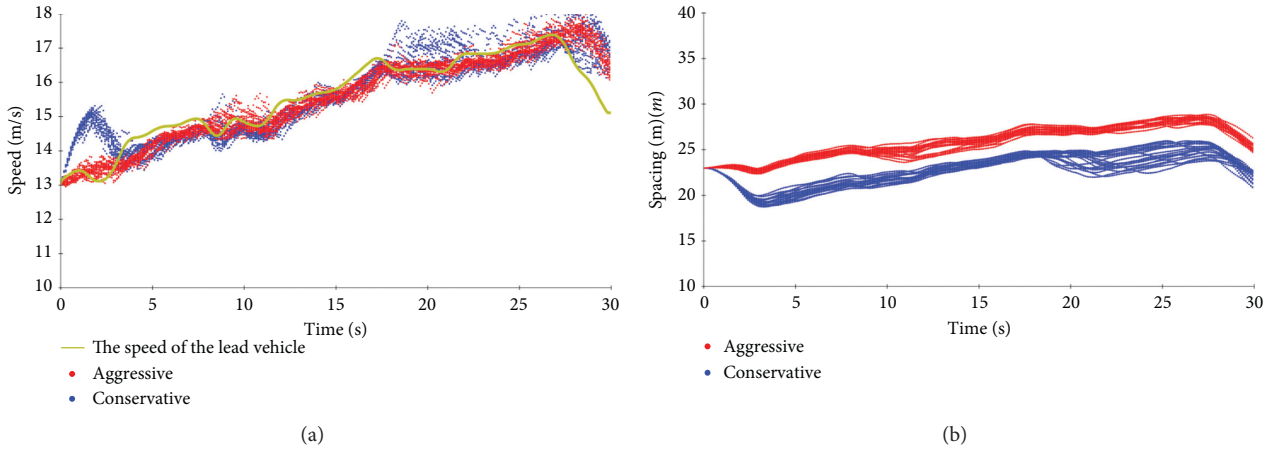


FIGURE 10: Comparison of the strategies of the two groups with different driving styles. (a) and (b) are the simulation results of speed and spacing, respectively.

loss function shown in equation (13) which minimizes the RMSPE of speed and spacing.

$$L(w,b) = \frac{\left(h_n^{\text{simu}}(t) - h_n^{\text{obs}}(t)\right)^2}{\left(h_n^{\text{obs}}(t)\right)^2} + \frac{\left(v_n^{\text{simu}}(t) - v_n^{\text{obs}}(t)\right)^2}{\left(v_n^{\text{obs}}(t)\right)^2} \tag{13}$$

where w, b denotes the weights and bias in the RNN model, $h_n^{\text{obs}}(t), h_n^{\text{obs}}(t)$ are the speed and spacing at time t in the observed nth trajectory, and $v_n^{\text{simu}}(t), s_n^{\text{simu}}(t)$ are the simulated speed and spacing at time t for the nth trajectory.

6. Results

6.1. Performance Comparison. The average performances of the three models in the fivefold cross-validation tests using the data from the aggressive and conservative groups were compared in this section. Tables 3 and 4 present the results on the training sets and the test sets, respectively. The DIRL had the lowest RMSPE of spacing and MHD_{50} in both the training sets and the test sets. Although the NN and the RNN model had lower RMSPE of speed in the test sets, the overall error of the DIRL in reproducing drivers' trajectories was lower than that in the other two models. For the two kinds of BC models, RNN outperformed the NN model as it achieved lower RMSPE and MHD_{50} than the NN model.

Figure 8 presents the simulation results of speed and spacing for two car-following periods randomly selected from the datasets. As can be seen, the DIRL model tracks the empirical speed and spacing more closely than the other two models. The simulation results of speed for the NN and RNN model are smoother than those of the DIRL model because the former models output a continuous action, while the latter model outputs a discrete action.

6.2. The Learned Characteristics of the Model. Since the proposed model was trained with data from two groups of drivers with different driving styles, we expected that the learned models would exhibit features of both groups. Therefore, the learned value of the two driving styles, which represents the expected long-term return, is compared in this section. As depicted in Figure 9, the states with a higher value represent the preferable states, which drivers try to achieve during car-following. For the same distance to the lead vehicle, the aggressive drivers preferred a higher speed than the conservative drivers. The high-value area ($V \geq 0.8$, in red) for the aggressive drivers has a steeper slope as indicated by the angle θ between the black-dashed line and the x-axis. Since the cotangent of the angle θ is proportional to the value of TH, a larger angle means a shorter TH. Hence, the comparison of the angle θ in the two figures shows that the aggressive drivers favor a shorter TH. Besides, the width of the high-value area for the aggressive is wider compared with the conservative; it indicates that the aggressive drivers' preferred TH has a larger variance than that of the conservative drivers. This result is in good agreement with the details shown in the boxplot of TH for the two groups of drivers in Figure 4.

It is also found that the high-value region of the speed becomes wider with an increase in the spacing to the lead vehicle in the two figures. The interpretation is that when the spacing is small, drivers must control the speed more precisely to prevent colliding. As the distance increases, drivers have more flexibility for speed control.

The learned policies of the two groups were compared by assuming that both groups were following the same leader. The initial states of this car-following event and the speed of the leader were input from the collected data. The learned stochastic policy was run 20 times for both groups. As shown in Figure 10, the aggressive group (in blue) maintained a smaller distance compared to the conservative group (in red) during the simulation. Both the aggressive and conservative drivers accelerated to follow the leader. However, the aggressive drivers increased the speed more quickly in the first 4 s, resulting in less distance to the leader compared with the conservative drivers.

7. Discussion and Conclusion

In this study, we propose a car-following model based on Max-Ent DIRL. The proposed model learns the rewards of drivers during car-following which were approximated by an NN. The policy of drivers was solved by an RL algorithm of softmax version of value iteration. Tested on actual driving data, the results showed that the proposed model outperformed the BC models NN and RNN by providing the lowest RMSPE and MHD_{50} in replicating drivers' car-following trajectories. The better performance of the proposed model can be explained by the more general objective compared with the BC models. The DIRL model reproduces drivers' policy by firstly learning drivers' decision-making mechanisms (i.e., the rewards), whereas the BC approaches only learn the state-action relationships. Since the policy was solved by the RL algorithm that is based on the assumption of maximizing long-term rewards, the obtained policy then has the ability of long-term planning. In contrast, the BC methods do not include long-term planning in its model training objectives. The simulation results for the two car-following trajectories confirmed the superior ability of long-term planning for the DIRL model. The derivation between the simulated spacing and the empirical data for the BC models becomes lager as the simulation continues. On the contrary, the simulation error does not accumulate during the simulation for the DIRL model. Moreover, the better performance of the RNN model found in this study is in line with previous studies [17, 34]. Compared with the NN model that only relies on information in the current time step for predication, the advantage of using historical information makes the RNN model more suitable for time series prediction.

The present study also demonstrates that the proposed model could capture the characteristics of different driving styles of human drivers. The learned value and policy matched those of the drivers with distinct driving styles. The fully connected NN applied in this study was trained to capture the relevant features that represented the drivers' preferences or objectives in car-following scenarios.

The IRL method used in this study provides a new perspective to explain driver behavior and to model different driving strategies. However, solving the IRL problem is computationally expensive, which makes it challenging to apply to high-dimensional systems. Recent studies that have applied adversarial learning to IRL have shown an ability to scale the method to solve complex problems [22, 47]. Future studies should consider these new approaches.

The present study had some important limitations. First, the participants in the present study are all male, so a broader sample is needed in future research. Second, the proposed model does not consider drivers' reaction delay and memory effect for speed control during car-following. Future studies should take these factors into account.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was jointly supported by the National Key R&D Program of China under Grant 2019YFB1600500, the Changjiang Scholars and Innovative Research Team in University under Grant IRT_17R95, the National Natural Science Foundation of China (51775053 and 51908054), and the Fundamental Research Funds for the Central Universities (300102228506).

References

[1] R. Fu, Z. Li, Q. Sun, and C. Wang, "Human-like car-following model for autonomous vehicles considering the cut-in behavior of other vehicles in mixed traffic," *Accident Analysis & Prevention*, vol. 132, Article ID 105260, 2019.

[2] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.

[3] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.

[4] M. Brackstone and M. McDonald, "Car-following: a historical review," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 181–196, 1999.

[5] H. Peng, "Evaluation of driver assistance systems-a human centered approach," in *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, Hiroshima, Japan, September 2002.

[6] A. Sharma, Z. Zheng, A. Bhaskar, and M. M. Haque, "Modelling car-following behaviour of connected vehicles with a focus on driver compliance," *Transportation Research Part B: Methodological*, vol. 126, pp. 256–279, 2019.

[7] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143–163, 2016.

[8] A. Talebpour, H. S. Mahmassani, and F. E. Bustamante, "Modeling driver behavior in a connected environment: integrated microscopic simulation of traffic and mobile wireless telecommunication systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2560, no. 1, pp. 75–86, 2016.

[9] L. Ye and T. Yamamoto, "Modeling connected and autonomous vehicles in heterogeneous traffic flow," *Physica A: Statistical Mechanics and Its Applications*, vol. 490, pp. 269–277, 2018.

[10] L. A. Pipes, "An operational analysis of traffic dynamics," *Journal of Applied Physics*, vol. 24, no. 3, pp. 274–281, 1953.

[11] R. Jiang, Q. Wu, and Z. Zhu, "Full velocity difference model for a car-following theory," *Physical Review E*, vol. 64, no. 1, p. 17101, 2001.

[12] A. Kesting and M. Treiber, "Calibrating car-following models by using trajectory data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2088, no. 1, pp. 148–156, 2008.

[13] Z. He, L. Zheng, and W. Guan, "A simple nonparametric car-following model driven by field data," *Transportation Research Part B: Methodological*, vol. 80, pp. 185–201, 2015.

[14] V. Papathanasopoulou and C. Antoniou, "Towards data-driven car-following models," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 496–509, 2015.

[15] L. Chong, M. M. Abbas, and A. Medina, "Simulation of driver behavior with agent-based back-propagation neural network," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2249, no. 1, pp. 44–51, 2011.

[16] H. Jia, Z. Juan, and A. Ni, "Develop a car-following model using data collected by "five-wheel system,"" in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, Shanghai, China, October 2003.

[17] M. Zhou, X. Qu, and X. Li, "A recurrent neural network based microscopic car following model to predict traffic oscillation," *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 245–264, 2017.

[18] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, May 2010.

[19] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, Stanford, CA, USA, July 2000.

[20] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[21] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: a survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[22] A. Kuefler, "Imitating driver behavior with generative adversarial networks," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, June 2017.

[23] H. Gao, "Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making," *International Journal of Advanced Robotic Systems*, vol. 15, no. 6, 2018.

[24] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: studies in car following," *Operations Research*, vol. 6, no. 2, pp. 165–184, 1958.

[25] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, Cambridge, MA, USA, 2018.

[27] Z. Cao, C. Liu, M. Zhou, and R. Huang, "Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 825–837, 2019.

[28] A. Xi, T. W. Mudiyanselage, D. Tao, and C. Chen, "Balance control of a biped robot on a rotating platform based on efficient reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 938–951, 2019.

[29] T. Liu, B. Tian, Y. Ai, L. Li, D. Cao, and F.-Y. Wang, "Parrallel reinforcement learning: a framework and case study," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 4, pp. 827–835, 2018.

[30] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, Association for Computing Machinery, Banff, Canada, July 2004.

[31] B. D. Ziebart, "Maximum entropy inverse reinforcement learning," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)*, Chicago, IL, USA, July 2008.

[32] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," 2015, https://arxiv.org/abs/1507.04888.

[33] R. Hecht-Nielsen, "3-theory of the backpropagation neural network∗∗based on "nonindent" by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593–611, 1989," in *Neural Networks for Perception*, H. Wechsler, Ed., pp. 65–93, Academic Press, Cambridge, MA, USA, 1992.

[34] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 348–368, 2018.

[35] G. Lu, B. Cheng, Q. Lin, and Y. Wang, "Quantitative indicator of homeostatic risk perception in car following," *Safety Science*, vol. 50, no. 9, pp. 1898–1905, 2012.

[36] E. R. Boer, "Car following from the driver's perspective," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 201–206, 1999.

[37] J. Wang, C. Yu, S. E. Li, and L. Wang, "A forward collision warning algorithm with adaptation to driver behaviors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1157–1167, 2016.

[38] H.-H. Yang and H. Peng, "Development of an errorable car-following driver model," *Vehicle System Dynamics*, vol. 48, no. 6, pp. 751–773, 2010.

[39] T. Liu, "Car-following warning rules considering driving styles," *China Journal of Highway and Transport*, vol. 33, pp. 170–180, 2020.

[40] T. Liu and Selpi, "Comparison of car-following behavior in terms of safety indicators between China and Sweden," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3696–3705, 2020.

[41] M. Zhu, X. Wang, A. Tarko, and S. E. Fang, "Modeling car-following behavior on urban expressways in Shanghai: a naturalistic driving study," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 425–445, 2018.

[42] S. Aranganayagi and K. Thangavel, "Clustering categorical data using silhouette coefficient as a relocating measure," in *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, IEEE, Sivakasi, Tamil Nadu, January 2007.

[43] V. Punzo and M. Montanino, "Speed or spacing? Cumulative variables, and convolution of model errors and time in traffic flow models validation and calibration," *Transportation Research Part B: Methodological*, vol. 91, pp. 21–33, 2016.

[44] M. Shimosaka, T. Kaneko, and K. Nishi, "Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning," in *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, October 2014.

[45] M. Wulfmeier, D. Z. Wang, and I. Posner, "Scalable cost-function learning for path planning in urban environments," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, October 2016.

[46] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[47] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proceedings of the Advances in Neural Information Processing Systems*, Barcelona, Spain, December 2016.