Interaction-Aware Planning With Deep Inverse Reinforcement Learning for Human-Like Autonomous Driving in Merge Scenarios

Jiangfeng Nan, Weiwen Deng, *Member, IEEE*, Ruzheng Zhang, Ying Wang, Rui Zhao, and Juan Ding

***Abstract*—Merge scenarios on highway are often challenging for autonomous driving, due to its lack of sufficient tacit understanding on and subtle interaction with human drivers in the traffic flow. This, as a result, may impose serious safety risks, and often cause traffic jam with autonomous driving. Therefore, human-like autonomous driving becomes important, yet imperative. This article presents an interaction-aware decision-making and planning method for human-like autonomous driving in merge scenarios. Rather than directly mimicking human behavior, deep inverse reinforcement learning is employed to learn the human-used reward function for decision-making and planning from naturalistic driving data to enhance interpretability and generalizability. To consider the interaction factor, the reward function for planning is utilized to evaluate the joint trajectories of the autonomous driving vehicle (ADV) and traffic vehicles. In contrast to predicting trajectories of traffic vehicles with the fixed trajectory of ADV given by the upstream prediction model, the trajectories of traffic vehicles are predicted by responding to the ADV's behavior in this article. Additionally, the decision-making module is employed to reduce the solution space of planning via the selection of a proper gap for merging. Both the decision-making and planning algorithms follow a "sampling, evaluation, and selection" framework. After being verified through experiments, the results indicate that the planned trajectories with the presented method are highly similar to those of human drivers. Moreover, compared to the interaction-unaware planning method, the interaction-aware planning method behaves closer to human drivers.**

***Index Terms*—Deep inverse reinforcement learning, human-like, interaction, merge scenarios, planning.**

Manuscript received 23 May 2023; revised 27 June 2023; accepted 23 July 2023. Date of publication 26 July 2023; date of current version 23 February 2024. This work was supported in part by the Lingyan R&D Program of Zhejiang Province under Grant 2023C01238, and in part by the Jianbing R&D Program of Zhejiang Province under Grant 2023C01133. *(Corresponding author: Ruzheng Zhang.)*

Jiangfeng Nan and Rui Zhao are with the School of Transportation Science and Engineering, Beihang University, Beijing 100191, China (e-mail: nanjiangfeng@buaa.edu.cn; zhaor93@buaa.edu.cn).

Weiwen Deng is with the School of Transportation Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China (e-mail: wdeng@buaa.edu.cn).

Ruzheng Zhang is with the Horizon Robotics, Beijing 100094, China (e-mail: ruzheng01.zhang@horizon.ai).

Ying Wang is with the College of Computer Science and Technology, Jilin University, Jilin 130015, China (e-mail: wangying_jlu@163.com).

Juan Ding is with the PanoSim Technology Limited Company, Jiaxing 314000, China (e-mail: juan.ding@panosim.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3298912.

Digital Object Identifier 10.1109/TIV.2023.3298912

I. Introduction

While autonomous driving technology has developed rapidly, and traveling on complex interaction scenarios remains a major challenge for autonomous driving [1]. One of the most challenging scenarios for both ADV and the human driving vehicle is the merge scenario [2], [3]. Generally, the merging vehicle needs to select a proper gap on the target lane and complete the merging maneuver before the current lane ends. Due to strong interaction, ADV is expected to act like human drivers to make their behavior predictable for human-driving traffic vehicles [4], [5]. Additionally, human-like autonomous vehicles can reduce passenger takeover rates, especially in such challenging scenarios.

A. Decision Making and Planning in Merge Scenarios

There exists extensive literature on decision-making and planning in merge scenarios for autonomous driving. In earlier studies, rule-based methods [6], Markov processes [7], and utility theory models [8] are utilized to address decision-making and planning problems in merge scenarios. These traditional methods do not consider the interaction between the ego vehicle and other traffic vehicles, which goes against the driving habits of human drivers and is impolite and unsafe [9]. To reflect on the interaction, some studies predict the future trajectories of other traffic vehicles with the fixed trajectory of the ego vehicle. Deep neural networks, specifically recursive neural networks [10] and graph neural networks [11], are commonly employed for trajectory prediction. An integrated deep learning-based two-dimension trajectory prediction algorithm is proposed to predict integrated car-following and lane-changing behaviors [12]. The experimental results indicate this model can provide accurate short-term and long-term predictions. Additionally, multimodal trajectory prediction is exploited to predict the distribution of future trajectories for other traffic vehicles [13], [14], [15]. However, these open-loop prediction methods ignore the ego vehicle's influence on other traffic vehicles. In view of this, reinforcement learning (RL) is utilized to model interaction by setting up a training environment considering the ego vehicle's influence on other traffic vehicles [16], [17], [18]. Despite its potential, RL still faces major challenges in terms of interpretability and safety. Game theory is another popular approach to model interaction in lane-changing and merge scenarios [19], [20], [21], [22], [23]. To predict the trajectories of traffic vehicles, a “Leader-Follower Game Controller” is proposed in [24]. However, how to ensure the accuracy of hand-crafted reward functions of game participants in game theory is one of the main challenges [25]. Inverse reinforcement learning employed in this article can solve this challenge by learning the expert-used reward function from expert demonstrations.

B. Human-Like Autonomous Driving

If ADV exhibits human-like driving behavior, human-driven traffic vehicles will be easier to interact with them, and passengers will have more trust [26]. In recent years, studies on autonomous driving put more and more emphasis on human likeness. Algorithms for human-like autonomous driving include Behavior Cloning and Inverse Reinforcement Learning. A behavior cloning-based human-like lane-changing decision-making model is presented by combining the deep autoencoder network and XGBoost algorithm in [27]. Similarly, game theory-based behavior cloning is employed to mimic human lane-changing behavior in [28]. In contrast to behavior cloning, inverse reinforcement learning (IRL) does not mimic decision-making policy from human driving data directly. Instead, it learns the internal reward function that humans use for decision-making. IRL can often achieve better interpretability and generalizability, as the reward function inferred can explain the reasons for driver behavior. IRL has two branches: maximum margin [29] and maximum entropy [30]. The maximum entropy IRL has become more widely applied by reason that it can address the issue of reward function ambiguity. Many pieces of research have utilized feature-based IRL to learn human driving behavior from demonstrations [31], [32], [33], [34], [35]. To improve fitting ability, deep IRL (DIRL) utilizes deep neural networks to represent the driver’s reward function [36], [37]. However, it limits DIRL’s development in that solving DIRL is difficult owing to the large and continuous state and action spaces.

C. Contributions

In light of the challenges in merge scenarios, we present an interaction-aware decision-making and planning method for human-like autonomous driving based on deep inverse reinforcement learning (DIRL). This method consists of two stages: decision-making and planning. DIRL is employed to learn the human-used reward function for decision-making and planning from naturalistic driving data. The Q network is utilized to represent the human-used reward function for decision-making and the reward network is employed to fit the human-used reward function for planning.

Specifically, the planning algorithm follows a “sample, evaluate and select” framework. Firstly, the candidate trajectories of the ego vehicle are sampled and the future trajectories of other traffic vehicles are predicted. We employ the high-level driving intention-based trajectory sampling method to address the challenge of solving DIRL due to the large and continuous state and action spaces. Additionally, in contrast to most trajectory prediction models, the ego vehicle’s behavior is considered when the trajectories of traffic vehicles are predicted. Then the reward network for planning is utilized to evaluate the joint trajectory of the ego vehicle and traffic vehicles. Finally, the optimal trajectory is selected from the candidate trajectory set as the planning result.

The lane-changing process is often divided into decision-making and implementation [38]. Similarly, to reduce the sampling space in the planning stage, a DIRL-based decision-making module is employed to select the most suitable gap for merging before the planning stage. Most decision-making models for merge scenarios prefer waiting for the proper merging gap to appear by slowing down [39]. While this strategy is reasonable for short lanes, it is overly conservative and does not match human driving behavior in longer lanes. To merge into the target lane more efficiently, the active decision-making module is employed to find the proper merging gap, instead of passively waiting for it.

The main contributions and novelties of this article are listed as follows:

1) A sample-based DIRL is presented to plan human-like merging trajectories to address the challenge of solving DIRL caused by the large and continuous state and action spaces.
2) To account for interactions during trajectory planning, the joint trajectories of the ego vehicle and traffic vehicles are evaluated by the reward network for planning. In contrast to most trajectory prediction models, the ego vehicle’s behavior is considered when the trajectories of traffic vehicles are predicted by FIRL.
3) To reduce the solution space during the planning phase, a decision-making module is utilized to select the most appropriate merging gap before planning the merging trajectory. The Q network for decision-making is trained with naturalistic driving data. Therefore, its decision is more human likeness than the DQN with a hand-crafted reward function.

The rest of the article is summarized as follows. Section II introduces the decision-making and planning methods. Section III provides a detailed account of the training methods for the Q network corresponding to the decision-making stage, and the reward network for joint trajectories of ego vehicle and traffic vehicles in the planning stage. Section IV verifies the presented methods through experiments. Finally, a summary of this article is concluded.

II. Methodology

A. Problem Formulation

The merge scenario is illustrated in Fig. 1, where the ego vehicle in the acceleration lane needs to merge into the main lane and must complete the maneuver before reaching the end of the acceleration lane. To strike a balance between computational cost and performance, we focus on five surrounding traffic vehicles (V1, V2, V3, V4, and V5 as shown in Fig. 1), where V1, V2, V3, and V4 form three gaps. In order to maintain a consistent number of states while describing the merging scenario, virtual states are assigned to the non-existent vehicles. The principle behind assigning virtual states is to ensure that the presence of

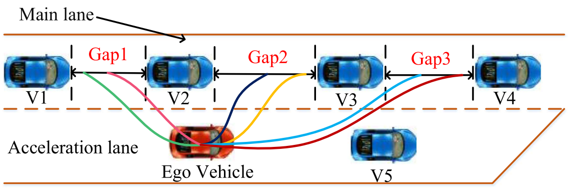


Fig. 1. Merge scenario.

virtual traffic vehicles does not impact the merging behavior of the ego vehicle. If V1 or V2 does not exist, their virtual relative distance and velocity are assigned to be -100 m and -30 m/s, respectively. If V3, V4, or V5 does not exist, their virtual relative distance and velocity are assigned to be 100 m and 30 m/s, respectively. Note that V2 and V3 represent the closest following and leading vehicles on the target lane with respect to the ego vehicle. If V2 is not present, V1 is absent as well. Similarly, the absence of V3 implies the absence of V4. Although there are many safe merging trajectories, driving safety is no longer the only requirement for the ego vehicle. In addition, the ego vehicle is expected to behave similarly to human drivers and interact with traffic vehicles in reason.

To clarify, the gap where the ego vehicle merges is referred to as the "target gap". When the ego vehicle merges into the main lane, it mainly affects the following vehicle of the target gap. As a result, the interaction between the ego vehicle and the following vehicle of the target gap is only considered.

Fig. 1 indicates that the solution space of the merge trajectory can be partitioned into three parts: the trajectory to merge into "Gap1", the trajectory to merge into "Gap2", and the trajectory to merge into "Gap3". To reduce the solution space of the planning trajectories, a decision-making module is first employed to select the most suitable merging gap as the target gap, and then the optimal trajectory is planned within the solution space of the target gap. In this way, the solution space of the planning algorithm can be reduced to one-third of its original size. The details of the decision-making and planning algorithms will be described in the following sections.

B. Sampling-Based Deep Inverse Reinforcement Learning

The difficulty of solving DIRL limits its development. To address this challenge, we present a sample-based DIRL method and derive the gradient formula. According to the maximum entropy principle, the probability of selecting a trajectory is proportional to the natural exponent of its reward, given by

$$P(\tau|\theta) = \frac{e^{R(\tau|\theta)}}{Z(\theta)}, \tag{1}$$

where $P(\tau|\theta)$ represents the probability of trajectory τ being chosen; $R(\tau|\theta)$ is the reward of trajectory τ; θ is the parameters of the reward function; $Z(\theta) = \int_D e^{R(\tau|\theta)} d\tau$ is referred to as the normalization function or partition function; D is the set of all possible trajectories that the agent can select.

The continuous and huge state space makes it difficult to compute the partition function in merging scenarios. The partition function is approximated by sampling candidate trajectories in the state space:

$$Z(\theta) \approx \sum_{\tau_i \in \phi} e^{R(\tau_i|\theta)}, \tag{2}$$

where ϕ denotes all sampled candidate trajectories.

Therefore, the probability of trajectory τ being chosen is

$$P(\tau|\theta) \approx \frac{e^{R(\tau|\theta)}}{\sum_{\tau_i \in \phi} e^{R(\tau_i|\theta)}}. \tag{3}$$

All trajectories in ϕ have the same initial state as trajectory τ.

The goal of maximum entropy inverse reinforcement learning is to maximize the likelihood of expert demonstration trajectories by tuning the parameters θ of the reward function:

$$\theta^* = \arg\max_{\theta} \sum_{\tau_e \in E} \log P(\tau_e|\theta), \tag{4}$$

where E represents all expert demonstration trajectories.

Therefore, the objective function of the IRL can be given by

$$\begin{aligned} J(\theta) &= \sum_{\tau_e \in E} \log P(\tau_e|\theta) \\ &= \sum_{\tau_e \in E} \log \frac{e^{R(\tau_e|\theta)}}{\sum_{\tau_i \in \phi_e} e^{R(\tau_i|\theta)}} \\ &= \sum_{\tau_e \in E} \left[R(\tau_e|\theta) - \log \sum_{\tau_i \in \phi_e} e^{R(\tau_i|\theta)} \right]. \end{aligned} \tag{5}$$

After derivation, the gradient of the reward function parameters is:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_{\tau_e \in E} \left[\frac{\partial R(\tau_e|\theta)}{\partial \theta} - \frac{\sum_{\tau_i \in \phi_e} \left[e^{R(\tau_i|\theta)} \cdot \frac{\partial R(\tau_i|\theta)}{\partial \theta} \right]}{\sum_{\tau_i \in \phi_e} e^{R(\tau_i|\theta)}} \right] \\ &= \sum_{\tau_e \in E} \left[\frac{\partial R(\tau_e|\theta)}{\partial \theta} - \sum_{\tau_i \in \phi_e} \left[P(\tau_i|\theta) \cdot \frac{\partial R(\tau_i|\theta)}{\partial \theta} \right] \right] \\ &= \sum_{\tau_e \in E} \left[\frac{\partial R(\tau_e|\theta)}{\partial \theta} - \mathbb{E}_{P(\tau_i|\theta)} \left(\frac{\partial R(\tau_i|\theta)}{\partial \theta} \right) \right]. \end{aligned} \tag{6}$$

C. Interaction-Aware Planning Method

This section focuses on planning human-like trajectories for merging into the target gap selected by the decision-making module. The decision-making module will be described in detail in Section II-E.

The presented planning method follows the framework of "sampling, evaluation, and selection", as displayed in Fig. 2. To clarify, the following vehicle of the target gap is referred to as FVTG, and the leading vehicle of the target gap is referred to as LVTG in this article.

Sampling: The candidate merging trajectories are sampled that satisfy safety constraints. And, the trajectory of the traffic vehicle corresponding to each candidate trajectory is predicted. When sampling candidate trajectories, a "high-level driving intention"-based trajectory representation method is firstly employed to reduce the dimensionality of the merging

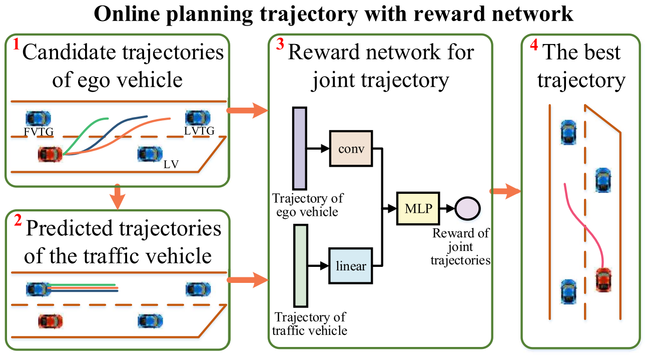

Online planning trajectory with reward network
1 Candidate trajectories of ego vehicle
FVTG
LVTG
LV
2 Predicted trajectories of the traffic vehicle
3 Reward network for joint trajectory
conv
Trajectory of ego vehicle
MLP
Reward of joint trajectories
linear
Trajectory of traffic vehicle
4 The best trajectory

Fig. 2. Planning algorithm follows a "sampling, evaluation, and selection" framework. Firstly, the candidate trajectories of the ego vehicle are sampled (as illustrated in the box 1) and the trajectory of the traffic vehicle corresponding to each candidate trajectory is predicted (as shown in the box 2). Then, as the box 3 shows, the reward network is utilized to evaluate the joint trajectory of the ego vehicle and the traffic vehicle. Finally, the optimal trajectory is selected from candidate trajectories, as illustrated in the box 4.

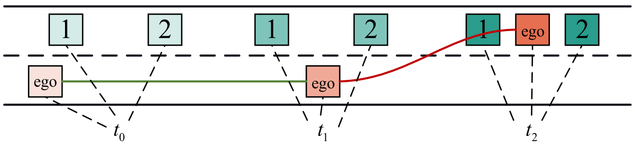


Fig. 3. Merging trajectory. The boxes represent vehicles at times t_0, t_1, and t_2. The number in the box denotes the vehicle ID. The orange box represents the ego vehicle, while the green box is the traffic vehicle.

trajectory, and then candidate trajectories are sampled in the lower-dimensional trajectory space. The specific trajectory representation method will be described in this section below.

Evaluation: The joint trajectories of the ego vehicle and the traffic vehicle are evaluated based on the reward network learned from naturalistic driving data using DIRL, which represents the human driver-used reward function for planning. The trajectory of the ego vehicle goes through a convolutional layer, while the trajectory of the traffic vehicle goes through a linear layer. After being concatenated, they are fed into the MLP layer and finally output joint reward. In merge scenarios, the considered traffic vehicle is FVTG. Unlike open-loop trajectory prediction models, the behavior of the ego vehicle is taken into account for predicting the trajectory of FVTG, as shown in the boxes "1" and "2" in Fig. 2. The prediction and evaluation methods for the trajectories of FVTG will be described in detail in Section II-D.

Selection: The trajectory of the ego vehicle with the highest reward is selected as the planning result.

The merging trajectory can be divided into two stages: the first stage is from t_0 to t_1 (represented by the green line) and the second stage is from t_1 to t_2 (denoted by the red line), as illustrated in Fig. 3. The first stage is the longitudinal adjustment stage, in which the ego vehicle adjusts its longitudinal position and velocity relative to the target gap. Then, the ego vehicle completes the lane-changing maneuver during the second stage.

In fact, the driving behavior of human drivers is governed by their high-level intentions. Compared to making decisions on low-level driving intentions (such as acceleration at each time step), making decisions on high-level driving intentions can greatly reduce the state space and make the trajectories smoother. Therefore, merging trajectories are generated by the high-level driving intentions. In the longitudinal adjustment, the high-level driving intentions consist of the ego vehicle's target longitudinal position $S_e(t_1)$ and velocity $v_e(t_1)$, as well as the required time $T_1 = t_1 - t_0$ for completing the longitudinal adjustment maneuver. The quintic polynomial is utilized to fit the ego vehicle's longitudinal position in the first stage in the S-L coordinate system, given by

$$S_e(t) = b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0. \tag{7}$$

So, the velocity and acceleration of the ego vehicle are respectively represented as:

$$v_e(t) = 5b_5 t^4 + 4b_4 t^3 + 3b_3 t^2 + 2b_2 t + b_1. \tag{8}$$

$$a_e(t) = 20b_5 t^3 + 12b_4 t^2 + 6b_3 t + 2b_2. \tag{9}$$

Given the initial state $S_e(t_0)$, $v_e(t_0)$, and $a_e(t_0)$, as well as the target state $S_e(t_1)$, $v_e(t_1)$ and $a_e(t_1) = 0$, the coefficients of the polynomial can be expressed as

$$\begin{cases} b_5 = 6\frac{S_e(t_1)-S_e(t_0)}{t_1^5} - 3\frac{v_e(t_1)+v_e(t_0)}{t_1^4} + \frac{a_e(t_1)-a_e(t_0)}{2t_1^3} \\ b_4 = 15\frac{S_e(t_0)-S_e(t_1)}{t_1^4} + \frac{7v_e(t_1)+8v_e(t_0)}{t_1^3} + \frac{3a_e(0)-2a_e(t_1)}{2t_1^2} \\ b_3 = 10\frac{S_e(t_1)-S_e(t_0)}{t_1^3} - \frac{4v_e(t_1)+6v_e(t_0)}{t_1^2} + \frac{a_e(t_1)-3a_e(t_0)}{2t_1} \\ b_2 = \frac{a_e(t_0)}{2} \\ b_1 = v_e(t_0) \\ b_0 = S_e(t_0) \end{cases}. \tag{10}$$

In the second stage, the quintic polynomial is also employed to fit the lateral trajectory of the lane change. The L-coordinate in the S-L coordinate system is expressed as

$$L(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0. \tag{11}$$

Given the boundary conditions $(L(t_0), \dot{L}(t_0), \ddot{L}(t_0), L(t_1), \dot{L}(t_1), \ddot{L}(t_1))$, the coefficients of the lateral trajectory polynomial for lane change can be determined.

The candidate trajectories can be generated by sampling high-level driving intentions $\{T_1, S_e(t_1), v_e(t_1)\}$. The sampling range of the required time T_1 is [0, 10] s with a sampling interval of 1 s. The sampling range of the target velocity $v_e(t_1)$ is $[v_g - 2, v_g + 2]$ m/s with a sampling interval of 0.5 m/s, where v_g is the velocity of the vehicle closest to the ego vehicle among FVTG and LVTG. The sampling range of the target longitudinal position $S_e(t_1)$ is $[S_{gf}, S_{gl}]$ m, with a sampling interval of 2 m, where S_{gf} and S_{gl} are the positions of FVTG and LVTG at time t_1, respectively. Any candidate trajectories resulting in collisions will be removed.

In order to improve the fitting ability of inverse reinforcement learning, the reward network is employed to represent the human-used reward function for planning. The architecture of the reward network is shown in Fig. 4. To interact with traffic vehicles in reason, the reward for the trajectory of FVTG is fed into the reward network together with the candidate trajectory of the ego vehicle. The merging trajectory can be represented by a matrix with two dimensions: time and state. Therefore, the merging trajectory can be fed into a convolutional neural

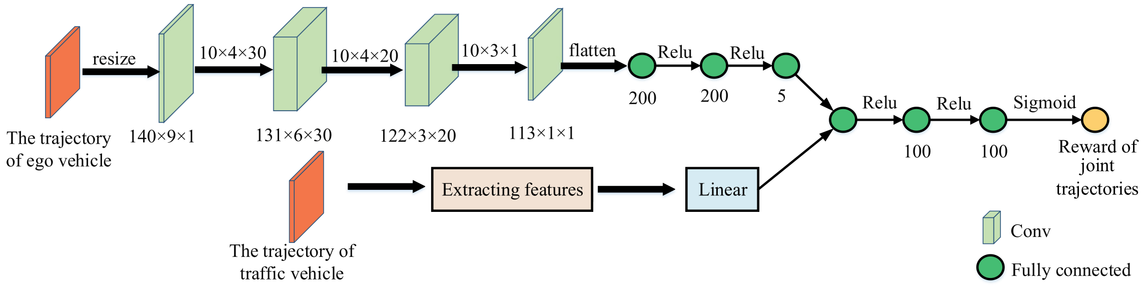


Fig. 4. Architecture of the reward network for joint trajectories of the ego vehicle and the traffic vehicle.

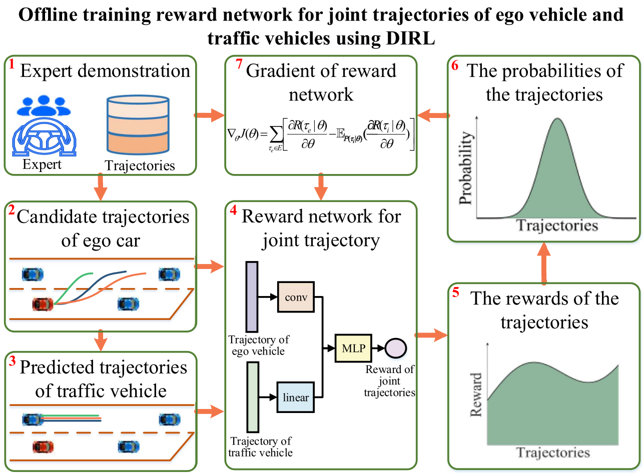

Offline training reward network for joint trajectories of ego vehicle and traffic vehicles using DIRL
1 Expert demonstration
Expert
Trajectories
7 Gradient of reward network
$\nabla_\theta J(\theta)=\sum_{\tau_e\in E}\left[\frac{\partial R(\tau_e|\theta)}{\partial\theta}-\mathbb{E}_{P(\tau_i|\theta)}\left(\frac{\partial R(\tau_i|\theta)}{\partial\theta}\right)\right]$
6 The probabilities of the trajectories
Probability
Trajectories
2 Candidate trajectories of ego car
4 Reward network for joint trajectory
conv
Trajectory of ego vehicle
MLP
Reward of joint trajectories
linear
Trajectory of traffic vehicle
5 The rewards of the trajectories
Reward
Trajectories
3 Predicted trajectories of traffic vehicle

Fig. 5. Framework of training reward network for joint trajectories of ego vehicle and traffic vehicles.

network to extract its features. The state of the ego vehicle trajectory consists of the ego vehicle's velocity, acceleration, and lateral position, as well as the relative distance and velocity with FVTG, LVTG, and the leading vehicle on the acceleration lane (V5). Because the trajectory time is different, the matrix representing the ego vehicle's trajectory needs to be resized before going through the convolutional network.

Sampled-based DIRL is employed to train the reward network. The framework of training reward network is illustrated in Fig. 5. First, the candidate trajectory set of the ego vehicle is sampled with the same initial state as the expert demonstration trajectory, as shown in the boxes "1" and "2" in Fig. 5. Then, the candidate trajectories of the ego vehicle and the reward for the future trajectory of FVTG are fed into the reward network, as shown in the boxes "2", "3", "4", and "5" in Fig. 5. According to the reward output by the reward network, the probability of the candidate trajectories being selected is obtained according to Formula (3), as shown in the boxes "6" and "7" in Fig. 5. Then, the gradient of the reward network is determined by Formula (8) according to the average reward of the candidate trajectories of the ego vehicle and the reward of the expert trajectory, as shown in the boxes "1", "7", and "8" in Fig. 5. Finally, the parameters of the reward network are updated using the gradient ascent algorithm, as shown in the boxes "8" and "5" in Fig. 5. In order to quickly converge, the parameters of the linear layer are determined through feature-based inverse reinforcement learning methods, and are not trained together with the parameters in the convolutional layer and MLP. The detailed training steps will be described in Section III-C.

D. Interaction

In merge scenarios, the ego vehicles are mainly aware of the influence of their behaviors on the following vehicle of the target gap (FVTG), or more specifically if they would impose additional inconvenience to FVTG. To explicitly quantify such influence, some researchers predict trajectories of FVTG by IDM and quantify the ego vehicle's influence on FVTG based on its speed loss [35]. However, the ability of IDM to model driving behavior is limited due to the small number of model parameters. Note that it is not comprehensive that only speed loss of FVTG is employed to quantify the ego vehicle's influence on FVTG.

Therefore, we utilize feature-based inverse reinforcement learning (FIRL) to predict the trajectories of FVTG. Specifically, FIRL learns the internal reward function of FVTG from driving data and finds the trajectory that maximizes the reward function as the predicted trajectory. The reason for utilizing FIRL to predict trajectories is that (1) the ability of FIRL to model driving behavior is stronger [35]; and (2) the reward function learned by FIRL can more comprehensively quantify the ego vehicle's influence on FVTG. In merge scenarios, the smaller the reward is, the worse the ego vehicle's influence on FVTG is.

The reward function learned by FIRL is a linear function composed of multiple driving features. These features involve driving efficiency, comfort, and safety, which is also why the reward function learned by FIRL can more comprehensively quantify the ego vehicle's influence on FVTG. In the following, we will provide a detailed description of the driving features and the training process of the reward function.

The merge scenario is regarded as a car-following scenario with the sudden change in the leading vehicle for FVTG. FVTG can identify the lane-changing intention of the ego vehicle and consider it as a new leading vehicle when the ego vehicle uses its turn signal or historical trajectory to signal a lane change. In this scenario, the driving features of FVTG are selected as follows.

1) *Acceleration*:

$$f_1(s) = acc^2. \tag{12}$$

2) *Relative velocity to the ego vehicle:*

$$f_2(s) = v_r^2. \tag{13}$$

3) *Relative distance to the ego vehicle:*

$$f_3(s) = (d - d_{des})^2, \tag{14}$$

where d_{des} denotes the expected relative distance to the ego vehicle. The distance between FVTG and LVTG is regarded as the expected relative distance (the original relative distance before the ego vehicle cuts in).

4) *Penalty for collision:*

$$f_4(s) = \left(\frac{1}{d}\right)^2. \tag{15}$$

In feature-based inverse reinforcement learning, the form of the reward function is given by

$$r = -(\theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \theta_4 f_4). \tag{16}$$

The training method for the reward function in feature-based inverse reinforcement learning is as follows:

1) Initialize the parameters of the reward function $\boldsymbol{\theta}^0$;
2) Compute the empirical feature vector averaged over all demonstrations $\tilde{\mathbf{f}} = \frac{1}{N} \sum_{\boldsymbol{\tau} \in E} \mathbf{f}_{\boldsymbol{\tau}}$;
3) Solve for the optimal trajectory $\boldsymbol{\tau}$ based on the current reward function;
4) Compute the features $\mathbf{f}_{\boldsymbol{\tau}}$ of the optimal trajectory;
5) Compute the gradient of the reward function $\mathbf{g} = \mathbf{f}_{\boldsymbol{\tau}} - \tilde{\mathbf{f}}$;
6) Update the parameters of the reward function according to the gradient $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\omega} \cdot \mathbf{g}$, where $\boldsymbol{\omega}$ is Learning rate;
7) Repeat from (3) until convergence.

Given the reward function, MPC is employed to plan the optimal trajectory of the FVTG.

E. Active Decision-Making Method

To reduce the sampling space in the planning stage, a decision-making module is employed to select the most suitable gap for merging before the planning stage. Compared to the passive decision-making methods waiting for the proper merging gap to appear by slowing down, we present an active decision-making method. Active decision-making algorithms also follow the "sampling, evaluation, and selection" framework, as shown in "Online making decision with Q network" in Fig. 6.

Sampling: The ego vehicle selects the target gap from the three nearby gaps, as shown in Fig. 1. If no suitable gap is found, the ego vehicle will either accelerate or decelerate to search for proper merging gaps. Therefore, the active decision-making method has five candidate actions: "merging into Gap1", "merging into Gap2", "merging into Gap3", "accelerating to search for target gap", and "decelerating to wait for target gap".

Evaluation: A Q network is employed to evaluate candidate actions, as shown in Fig. 7. Q(S, A) represents the total reward that can be obtained by taking action A in state S. The states fed into the Q network consist of the ego vehicle's velocity, acceleration, and distance to the end of the acceleration lane, as well as the relative distance and velocity of the ego vehicle to the five surrounding traffic vehicles.

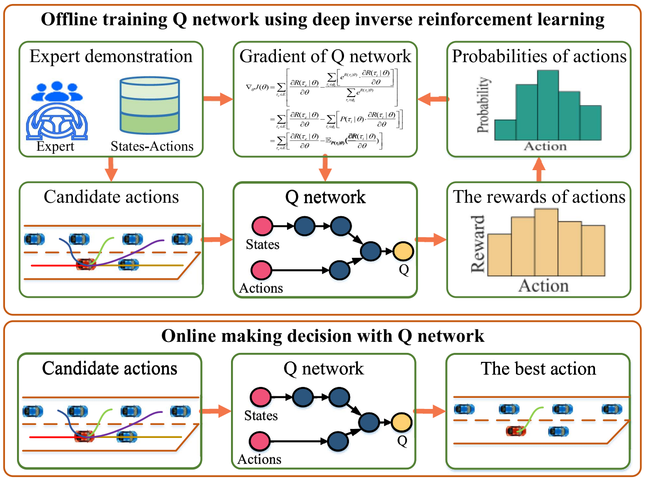


Fig. 6. Training and inferring framework of Q network based on DIRL.

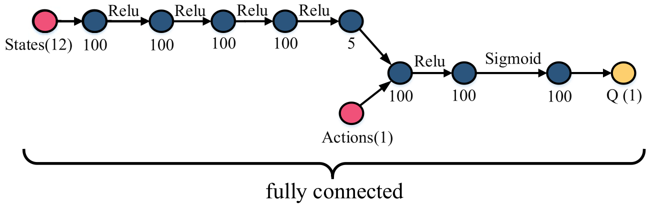


Fig. 7. Architecture of the Q network for decision-making.

Selection: The action with the highest Q value is selected as the optimal action.

During the offline training of the Q network using DIRL, as shown in Fig. 7, candidate actions are sampled for each driving state in the expert demonstration dataset. The goal of training the Q network is to maximize the likelihood described in Formula (7), which means making the actions selected by the Q network as consistent as possible with those selected by the expert in the same driving state. The gradient for training the Q network is given by Formula (6). The partition function described in Formula (2) is the sum of the natural exponentials of the rewards of the five candidate actions. Reinforcement learning (DQN) trains the Q network based on a hand-crafted reward function. In contrast, the Q network based on DIRL is trained with expert demonstration data. As a result, the Q network based on DIRL exhibits higher human likeness.

III. Training Networks

Both the Q network for decision-making and the reward network for planning are trained with naturalistic driving data in this section.

A. Data Preparation

The traffic data from 7:50–8:35 am on US Highway 101 in the Next Generation Simulation (NGSIM) dataset [40] is employed to train networks. The NGSIM dataset, collected by the United States Federal Highway Administration, is regarded as one of the largest publicly available naturalistic driving datasets and

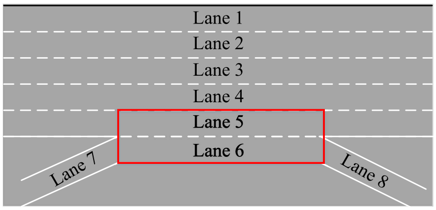


Fig. 8. Road structure of the NGSIM US101. The merging trajectories are in the red area.

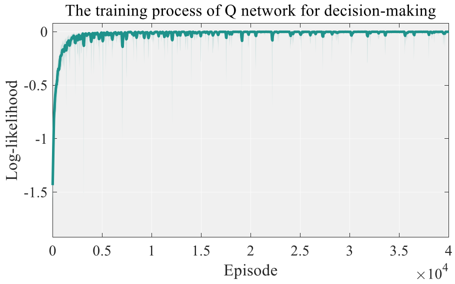


Fig. 9. Training process of the Q network.

has been widely used for research. In the NGSIM dataset, the sampling frequency of vehicle states is 10 Hz. The Savitzky-Golay filter is employed to smooth the NGSIM data with a 2-second window. As shown in Fig. 8, the recorded area of the NGSIM US101 dataset consists of five main lanes (Lane 1–5), one acceleration lane (Lane 6), and two auxiliary lanes (Lane 7 and Lane 8). The merging trajectories are in the red area in Fig. 8, where the merging vehicles on the acceleration lane first select the proper gap and then merge into the main lane. We extract 143 merging trajectories from the NGSIM US101 dataset.

B. Training Q Network for Decision Making

Fig. 6 lays out the framework of training Q network for decision-making, and the detailed training process is described in "Algorithm 1". The Adam optimizer is employed to train Q network with a learning rate of 0.0005 and the mini-batch size is set to 64. As the training proceeds, the log-likelihood of expert actions gradually increases and converges, as shown in Fig. 9.

We define the accuracy of the decision-making model as the average recall of 5 decision-making actions. To reflect the accuracy of the presented decision-making model, the Softmax-based classification model in supervised learning is employed as the baseline. The merging trajectories are divided into the training set and testing set to verify the model's generalizability across different trajectories. Three comparative experiments are conducted, where the ratio of the training dataset and test dataset is 9:1, 8:2, and 7:3, respectively. Table I illustrates that the DIRL-based decision-making algorithm achieves higher accuracy than the supervised learning-based algorithm.

TABLE I
ACCURACY OF THE DECISION-MAKING MODELS

The ratio of the training set to the testing set	9:1	8:2	7:3
Supervised learning	94.76%	66.33%	55.28%
DIRL (Ours)	**94.94%**	**76.41%**	**67.82%**

Algorithm 1: Training Q Network for Making Decision.

```
Input: Expert decision-making behavior dataset $E_1$
Output: Q network for making decision
Initialize the parameters of Q network $\theta_Q$
for episode $\leftarrow$ 1 to $M_1$ do
    Randomly select a subset $\kappa$ from $E_1$ as mini-batch
    for sample (driving state $s_i$ , expert action $A_e$) in $\kappa$ do
        for the action $A_j$ in 5 candidate actions do
            Calculate the Q value $Q(A_j|\theta_Q, s_i)$ of the action $A_j$ by inputting state $s_i$ and action $A_j$ into Q network
        end for
        for the action $A_j$ in 5 candidate actions do
            Calculate the probability of selecting action $A_j$ in driving state $s_i$ by formula $P(A_j|\theta_Q, s_i) = \frac{e^{Q(A_j|\theta_Q,s_i)}}{\sum_{j=1}^{5} e^{Q(A_j|\theta_Q,s_i)}}$
        end for
        Calculate average Q value $\mathbb{E}_{P(A_j|\theta_Q,s_i)}[Q(A_j|\theta_Q, s_i)]$ in driving state $s_i$
        Get the Q value $Q(A_e|\theta_Q, s_i)$ of the action $A_j$ taken by the expert in the driving state $s_i$
    end for
    $y = \sum_{\tau_e^i \in \kappa} (Q(A_e|\theta_Q, s_i) - \mathbb{E}_{P(A_j|\theta_Q,s_i)}[Q(A_j|\theta_Q, s_i)])$
    Calculate the gradient of the parameters of Q network by back propagation algorithm according $y$
    Update the parameters of Q network by the gradient ascent algorithm
end for
Output Q network for making decision
```

C. Training Reward Network for Planning

The framework of training the reward network for planning is shown in Fig. 5. "Algorithm 2" provides a detailed training process of the reward network. The Adam optimizer with the learning rate of 0.0001 is utilized to train the reward network and the mini-batch size is set as 32. Fig. 10 illustrates that the log-likelihood of expert trajectories gradually increases and converges with the increasing of episodes.

IV. EXPERIMENTS

A. Evaluation

The average log-likelihood of the expert demonstrated trajectories and the final displacement error (FDE) are selected as evaluation metrics of the model. Since our model is a probabilistic model, we define the final displacement error as the expected value of the final displacement error of all candidate trajectories. The ratio of the training dataset and test dataset is 9:1. The training process is shown in Fig. 11, which plots the

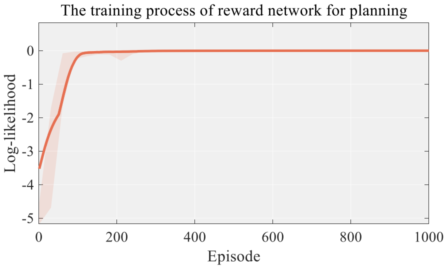


Fig. 10. Training process of the reward network for joint trajectories of ego vehicle and traffic vehicles.

Algorithm 2: Training Reward Network for Planning.

```
Input: Expert demonstration trajectory dataset $E_2$
Output: Reward network
Initialize the parameters of reward network $\theta_R$
for episode $\leftarrow$ 1 to $M_2$ do
    Randomly select a subset $\kappa$ from $E_2$ as mini-batch
    for expert demonstration trajectory $\tau_e^i$ in $\kappa$ do
        Determine the sampling space $\chi^i$ according to the
        initial state of expert demonstration trajectory $\tau_e^i$;
        Generate a candidate trajectory set $\phi^i$ with the
        same initial state as $\tau_e^i$ according to $\chi^i$, and $\tau_e^i$; is
        put into $\phi^i$
        Calculate the reward $R(\tau_e^i|\theta_Q)$ of $\tau_e^i$ by reward
        network
        for candidate trajectory $\tau_j^i$ in $\phi^i$ do
            Calculate the reward $R(\tau_j^i|\theta_Q)$ of $\tau_j^i$ by reward
            network
        end for
        Calculate $\tau_j^i$ distribution probability
        $P(\tau_j^i|\theta_Q) \approx \frac{e^{R(\tau_j^i|\theta_Q)}}{\sum_{\tau \in \phi^i} e^{R(\tau|\theta_Q)}}$
        Calculate the average reward
        $\mathbb{E}_{P(\tau_j^i|\theta_Q)}[R(\tau_j^i|\theta_Q)]$ of trajectories in $\phi^i$
    end for
    $y = \sum_{\tau_e^i \in \kappa} (R(\tau_e^i|\theta_Q) - \mathbb{E}_{P(\tau_j^i|\theta_Q)}[R(\tau_j^i|\theta_Q)])$
    Calculate the gradient of the parameters of reward
    network by back propagation algorithm according y
    Update the parameters of reward network by the
    gradient ascent algorithm
end for
Output reward network for planning
```

curves of the average log-likelihood and average FDE of the expert demonstration trajectories in the test set.

As seen in Fig. 11(a), the average log-likelihood of the expert demonstrated trajectories gradually increases and converges. The average log-likelihood increases to -0.25 from -4.81 at the beginning, which means that the probability of expert demonstrated trajectories being selected has increased from 0.81% to 77.88% after training. This gives rise to the decrease in the average FDE of the expert demonstration trajectories as shown in Fig. 11(b), which converges to 0.82 m. The experimental results demonstrate that the model converges to a stable value.

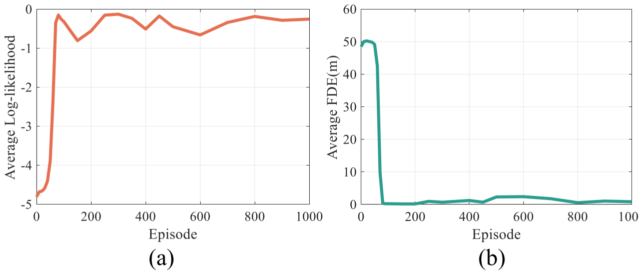


Fig. 11. Training process: (a) plot of the average log-likelihood of the expert demonstration trajectories; (b) plot of the average FDE of the expert demonstration trajectories.

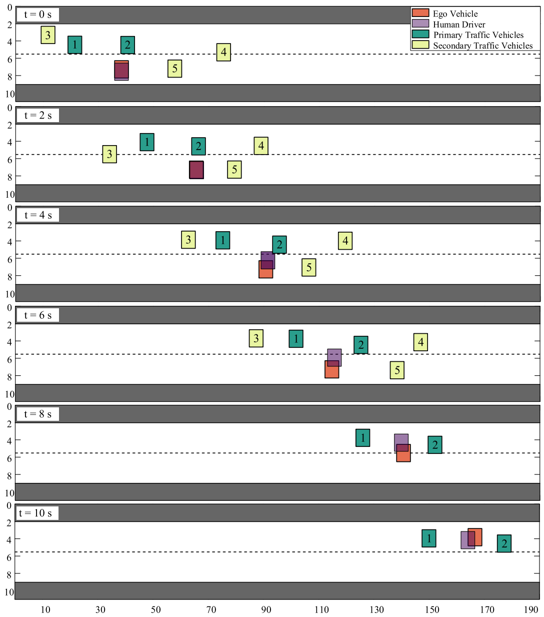


Fig. 12. Case study 1: merging into the gap by decelerating. The orange box represents the ego vehicle controlled by the presented model. The purple box is the position of the ego vehicle present in the US101 traffic dataset, which is controlled by the human driver. The primary traffic vehicles denoted by the blue boxes make up the target gap. Other traffic vehicles, called secondary traffic vehicles, are represented by faint yellow boxes. All traffic vehicles follow their original trajectories in the US101 traffic dataset. When the traffic vehicle leaves the ego vehicle's view, it will be removed from the figure.

B. Case Study

In this section, three cases from the test set are tested to verify the feasibility and effectiveness of the presented model intuitively. In each case, the ego vehicle is controlled by the presented method, while other traffic vehicles follow their original trajectories from the NGSIM. The above testing method is called the log-sim testing method, which is widely used in both academia and industry [24], [41], [42], as it can compare the decisions with those of the human drivers. All the driving scenarios are established and implemented on the Python platform.

Fig. 12 shows a scenario of the ego vehicle merging into the target gap by decelerating. At t = 0 s, the ego vehicle is very

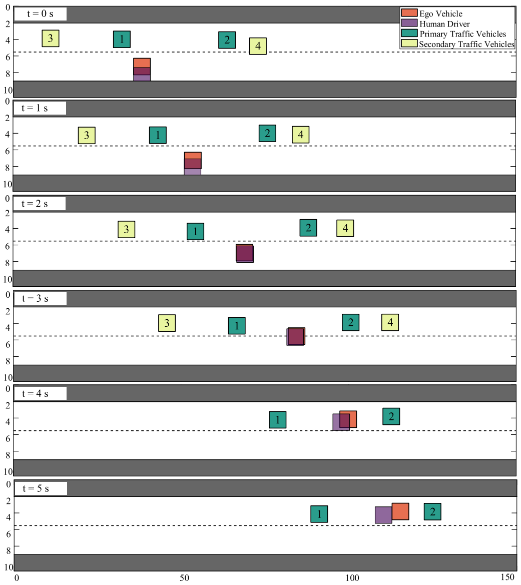


Fig. 13. Case study 2: merging into the gap by accelerating.

close to traffic vehicle 2, so it cannot merge into the target gap safely. Afterward, the ego vehicle slows down first to increase the relative distance from vehicle 2. Finally, the ego vehicle merges into the target gap with proper relative distance from vehicles 1 and 2. During the merging process, the trajectory of the ego vehicle controlled by the model is very close to that of the human driver. And, the FDE is only 2.56 m.

Fig. 13 shows a scenario of the ego vehicle merging into the target gap by accelerating. Firstly, the decision-making module selects the most suitable target gap. Since the distance between the ego vehicle and the traffic vehicle 1 is relatively close at t = 0 s, it will have a significant influence on the traffic vehicle 1 that the ego vehicle merges into the target gap immediately. Consequently, the ego vehicle accelerates to increase the distance from traffic vehicle 1 to merge into the target gap more safely. The FDE of the ego vehicle and the human driver is 5.06 m. During the process of merging into the gap by accelerating, the behavior of the ego vehicle controlled by the model is very similar to that of the human driver.

Fig. 14 shows a scenario of the ego vehicle merging into the target gap immediately. The ego vehicle is located in the middle of the target gap and the relative distance from vehicles 1 and 2 is proper at t = 0 s. Both the ego vehicle controlled by the model and the human driver merge into the target gap immediately without excessive adjustment of its relative longitudinal position to the target gap. The FDE of the ego vehicle and the human driver is 1.71 m. The results indicate that the presented model decidedly merges into the target gap like the human driver as soon as the relative position between the ego vehicle and the target gap is proper.

C. Effects of the Interaction Factor

The influence on other vehicles caused by the ego vehicle's trajectories is considered as the interaction factor. As described in Section II-D, the reward function learned by FIRL is employed to quantify the ego vehicle's influence on FVTG. The ego vehicle's influence on FVTG is the output of the linear layer in the reward network. In order to investigate the effect of the interaction factor on the model's accuracy, the average FDE and average likelihood of interaction-aware and interaction-unaware models are compared. In contrast to the interaction-aware model, the interaction-unaware model is achieved by removing the interaction factor from the reward network. The expert demonstration trajectories in the test set are selected and the results are shown in Table II.

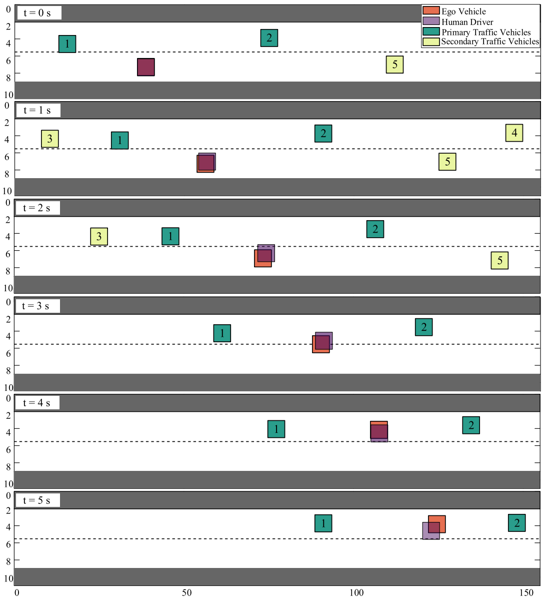


Fig. 14. Case study 3: merging into the gap immediately.

TABLE II
Effect of the Interaction Factor on Accuracy

	Average FDE (m)	Average likelihood
Interaction-Aware	**0.82**	**77.88%**
Interaction-Unaware	1.27	71.08%

It is apparent from Table II that removing the interaction factor in the reward network would impair the accuracy of the model, which suggests that the interaction factor is of importance in planning human-like trajectories. This is because the interaction factor is an important driving feature. Each human driver considers the influence on other traffic vehicles of their driving behavior for safety and courtesy. The interaction-aware model is more in line with this fact. Therefore, the behavior of the ego vehicle controlled by the interaction-aware model is more similar to that of the human driver than the interaction-unaware model.

The results in relation to safety are presented in Table III. The safety metrics considered are the minimum distance and minimum Time-to-Collision (TTC) between the ego vehicle and the FVTG. The results demonstrate that the interaction-unaware algorithm yielded a minimum distance of 0.25 m and a minimum

TABLE III
EFFECT OF THE INTERACTION FACTOR ON SAFETY

	Min Distance (m)	Min TTC (s)
Interaction-Aware	**8.65**	**13.05**
Interaction-Unaware	0.25	0.06

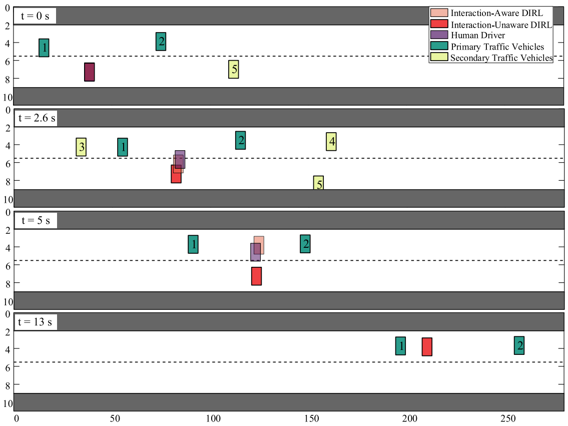
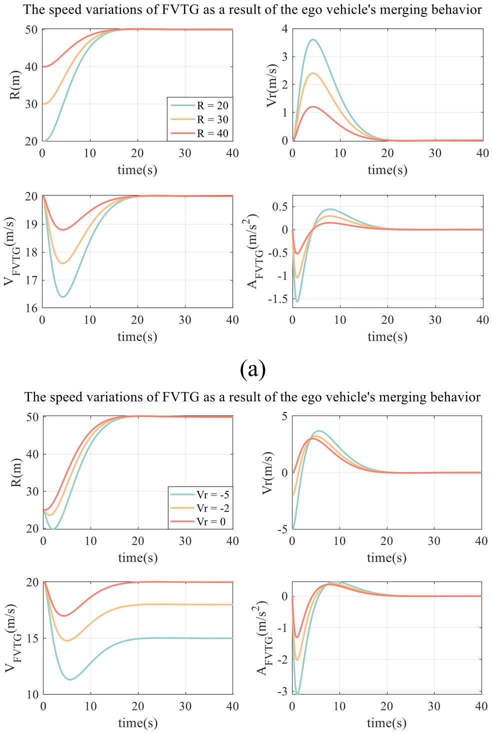


Fig. 15. Performance of interaction-aware and interaction-unaware models in the same merging scenario. The orange box represents the ego vehicle controlled by the interaction-aware model. The red box is the ego vehicle controlled by the interaction-unaware model. The purple box is the position of the ego vehicle present in the US101 traffic dataset, which is controlled by the human driver. Both the interaction-aware model and the human driver have completed the merging maneuver at t = 5 s, so they are removed from the figure at t = 13 s.



(a)

(b)

Fig. 16. Speed variations of FVTG as a result of the ego vehicle's merging behavior. Vr and R represent the relative velocity and distance between the ego vehicle and FVTG respectively; VFVTG and AFVTG are the velocity and acceleration of FVTG.

TTC of 0.06 s. In contrast, the interaction-aware algorithm resulted in a minimum distance of 8.65 m and a minimum TTC of 13.05 s. These results provide clear evidence of the superior safety performance of the proposed interaction-aware algorithm.

To provide a clear and concise comparison between interaction-aware and interaction-unaware models, we test both models in the same merging scenario and the results are illustrated in Fig. 15. The positions of the ego vehicles controlled by the interaction-aware model, interaction-unaware model, and the human driver overlap at t = 0 s. At t = 2 s, both the human driver and interaction-aware model initiate the merging maneuver, while the interaction-unaware model has not yet done so. At t = 5 s, both the human driver and the interaction-aware model have completed the merging maneuver, while the interaction-unaware model does not complete the merging maneuver until t = 13 s. The results suggest that removing the interaction factor in the reward network makes the planning model cannot accurately evaluate the influence of the ego vehicle on the traffic vehicle 1, causing the ego vehicle takes longer to complete the merging maneuver and is closer to the traffic vehicle 1.

In the merge scenario, the FVTG will decelerate to avoid collision when encountering merging vehicles. In a merge scenario, the speed variations of FVTG corresponding to different merging behaviors of the ego vehicle are depicted in Fig. 16. The speed variations of the FVTG in response to the ego vehicle's merging behavior with different initial relative distances at the same initial relative velocity are depicted in Fig. 16(a). For initial relative distances of 40 m, 30 m, and 20 m, the quantified values representing the ego vehicle's influence on the FVTG are 3.73, 7.44, and 11.15, respectively. This indicates that as the initial relative distance decreases, the ego vehicle's influence on the FVTG becomes significant. Fig. 16(b) displays the speed variations of the FVTG as a result of the ego vehicle's merging behavior with different relative velocities while maintaining the same relative distance. For initial relative velocities of 0 m/s, −2 m/s, and −5 m/s, the quantified values representing the ego vehicle's influence on the FVTG are 9.29, 12.06, and 18.14, respectively. This implies that as the initial relative velocities decrease, the ego vehicle's influence on the FVTG becomes intensified.

It can be seen from Fig. 17 that the smaller the relative distance between the ego vehicle and FVTG is, the greater the ego vehicle's influence on FVTG is. If the velocity of the ego vehicle is less than the velocity of FVTG, the ego vehicle's influence increases rapidly with a decrease in relative velocity. On the other hand, if the velocity of the ego vehicle is greater than the velocity of FVTG, the ego vehicle's influence slowly increases with the increase of relative velocity. The results illustrate that the ego vehicle's influence on FVTG is congenial with common sense. It is apparent from Fig. 18 that the joint reward gradually decreases with the increase of the ego vehicle's influence on FVTG. The reward network learns the fact that human drivers always try to reduce their influence on other traffic vehicles while achieving their driving goals, which is achieved by DIRL with expert demonstration data, rather than supervised learning

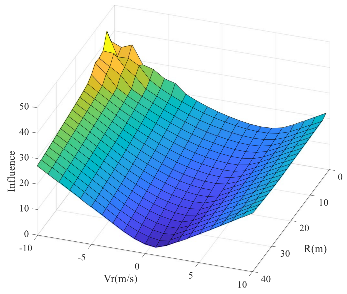


Fig. 17. Influence on FVTG caused by the ego vehicle's trajectories. The reward function learned by FIRL is employed to quantify the ego vehicle's influence on FVTG. The ego vehicle's influence on FVTG is the output of the linear layer in the reward network. Vr represents the relative velocity between the ego vehicle and FVTG at the moment of initiating merging; R is the relative distance between the ego vehicle and FVTG.

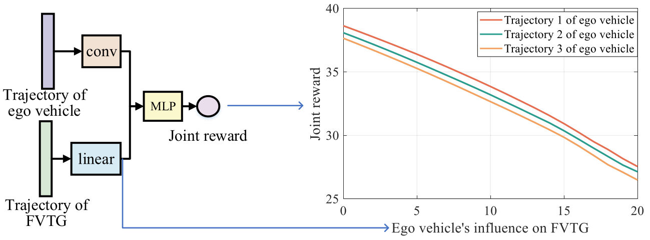


Fig. 18. Variation trend of joint reward with the change of the ego vehicle's influence on FVTG. The ego vehicle's influence on FVTG is the output of the linear layer in the reward network; Joint reward is the output of the reward network. Three ego vehicle trajectories are randomly selected to investigate the variation trend of joint reward with the change of the ego vehicle's influence on FVTG.

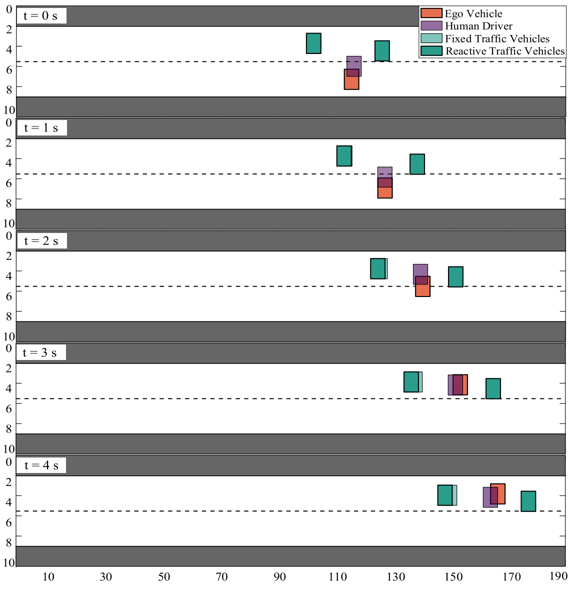


Fig. 19. FVTG reacts to the behavior of the ego vehicle.

with labeled data. The results demonstrate that interaction-aware DIRL can learn how the interaction factor affects the reward function utilized by human drivers for planning from expert demonstration data.

While it is common to compare behaviors with those of human drivers using recorded traffic data, we conduct an additional experiment to provide stronger evidence of the effectiveness of the proposed method. In this experiment, the FVTG is no longer fixed but controlled by FIRL. As shown in Fig. 19, the reactive trajectory of the FVTG closely aligns with the fixed trajectory, and the ego vehicle merges into the target gap safely.

V. Conclusion

This article presents an interaction-aware decision-making and planning method for merge scenarios. The method consists of two stages: decision-making and planning. The planning module follows a "sampling, evaluation, and selection" framework. Firstly, the candidate trajectories of the ego vehicle are sampled and the future trajectories of FVTG are predicted by considering the trajectory of the ego vehicle. To address the challenge of solving DIRL caused by the large and continuous state and action spaces, a high-level driving intention-based trajectory sampling method is presented in merge scenarios. Then, the candidate trajectories of the ego vehicle and the future trajectories of the FVTG are evaluated using the reward network trained by DIRL. Finally, the optimal trajectory is selected from the candidate trajectory set as the planning result. The trajectories of the FVTG are predicted while taking into account the influence of the ego vehicle on it, in contrast to most motion prediction models which ignore such influence. In the decision-making stage, the decision-making module selects the most suitable merging gap as the target gap, and the solution space of the planning algorithm can be reduced to one-third of its original size accordingly. The Q network for decision-making is trained using offline driving data from experts with DIRL. Compared to DQN relying on a hand-crafted one-step reward function, the Q network based on DIRL exhibits higher human likeness. Experimental results show that the planned trajectories with the presented method are highly similar to those of human drivers. Moreover, compared to the interaction-unaware planning method, interaction-aware DIRL can learn how the interaction factor affects the reward function utilized by human drivers for planning. This, as a result, the interaction-aware planning method behaves closer to human drivers. In future work, we will try to apply the proposed method to more complex scenarios, such as intersections with stronger interaction.

References

[1] T. Ersal et al., "Connected and automated road vehicles: State of the art and future challenges," *Veh. Syst. Dyn.*, vol. 58, no. 5, pp. 672–704, 2020.

[2] X. Hu and J. Sun, "Trajectory optimization of connected and autonomous vehicles at a multilane freeway merging area," *Transp. Res. Part C: Emerg. Technol.*, vol. 101, pp. 111–125, Apr. 2019.

[3] C. Wei, Y. He, H. Tian, and Y. Lv, "Game theoretic merging behavior control for autonomous vehicle at highway on-ramp," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21127–21136, Nov. 2022.

[4] Z. Zhao et al., "Personalized car following for autonomous driving with inverse reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2891–2897.

[5] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2076–2087, Apr. 2021, doi: 10.1109/TITS.2020.3036984.

[6] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transp. Res. Part B: Methodological*, vol. 20, no. 5, pp. 403–414, Oct. 1986.

[7] S. Ulbrich and M. Maurer, “Probabilistic online POMDP decision making for lane changes in fully automated driving,” in *Proc. IEEE 16th Int. Conf. Intell. Transp. Syst.*, 2013, pp. 2063–2067.

[8] T. Toledo, “Modeling integrated lane-change behavior,” *Transp. Res. Rec.*, vol. 1857, no. 1, pp. 30–38, 2003.

[9] Q. Zhang, R. Langari, H. E. Tseng, D. Filev, S. Szwabowski, and S. Coskun, “A game theoretic model predictive controller with aggressiveness estimation for mandatory lane change,” *IEEE Trans. Intell. Veh.*, vol. 5, no. 1, pp. 75–89, Mar. 2020, doi: 10.1109/TIV.2019.2955367.

[10] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit .*, 2016, pp. 961–971.

[11] X. Li, X. Ying, and M. C. Chuah, “GRIP: Graph-based interaction-aware trajectory prediction,” in *Proc. IEEE Intell. Transp. Syst. Conf .*, 2019, pp. 3960–3966.

[12] K. Shi et al., “An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network,” *Physica A: Stat. Mechanics Appl.*, vol. 599, 2022, Art. no. 127303.

[13] H. Cui et al., “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 2090–2096.

[14] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, “MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction,” in *Proc. Conf. Robot Learn., Mach. Learn. Res.*, 2020, pp. 86–99.

[15] M. Liang et al., “Learning lane graph representations for motion forecasting,” in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 541–556.

[16] H. Li, N. Li, I. Kolmanovsky, and A. Girard, “Energy-efficient autonomous vehicle control using reinforcement learning and interactive traffic simulations,” in *Proc. IEEE Amer. Control Conf.*, 2020, pp. 3029–3034.

[17] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, “Driving in dense traffic with model-free reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 5385–5392.

[18] G. Li et al., “Deep reinforcement learning enabled decision-making for autonomous driving at intersections,” *Automot. Innov.*, vol. 3, no. 4, pp. 374–385, 2020.

[19] J. Nan, W. Deng, and B. Zheng, “Intention prediction and mixed strategy nash equilibrium-based decision-making framework for autonomous driving in uncontrolled intersection,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10316–10326, Oct. 2022.

[20] H. Kita, “A merging–giveway interaction model of cars in a merging section: A game theoretic analysis,” *Transp. Res. Part A: Policy Pract.*, vol. 33, no. 3/4, pp. 305–312, 1999.

[21] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, “Planning for autonomous cars that leverage effects on human actions,” in *Proc. Robot., Sci. Syst.*, Ann Arbor, MI, USA, 2016, vol. 2, pp. 1–9.

[22] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, “Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems,” *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1782–1797, Sep. 2018.

[23] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, “Hierarchical game-theoretic planning for autonomous vehicles,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9590–9596.

[24] K. Liu, N. Li, H. E. Tseng, I. Kolmanovsky, and A. Girard, “Interaction-aware trajectory prediction and planning for autonomous vehicles in forced merge scenarios,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 474–488, Jan. 2023.

[25] C. Wei, Y. He, H. Tian, and Y. Lv, “Game theoretic merging behavior control for autonomous vehicle at highway on-ramp,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21127–21136, Nov. 2022.

[26] A. Li, H. Jiang, J. Zhou, and X. Zhou, “Learning human-like trajectory planning on urban two-lane curved roads from experienced drivers,” *IEEE Access*, vol. 7, pp. 65828–65838, 2019.

[27] X. Gu, Y. Han, and J. Yu, “A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and XGBoost,” *IEEE Access*, vol. 8, pp. 9846–9863, 2020.

[28] H. Yu, H. E. Tseng, and R. Langari, “A human-like game theory-based controller for automatic lane changing,” *Transp. Res. Part C: Emerg. Technol.*, vol. 88, pp. 140–158, Mar. 2018.

[29] D. Silver, J. A. Bagnell, and A. Stentz, “Learning autonomous driving styles and maneuvers from expert demonstration,” in *Experimental Robotics*. Berlin, Germany: Springer, 2013, pp. 371–386.

[30] B. D. Ziebart et al., “Maximum entropy inverse reinforcement learning,” *AAAI*, vol. 8, pp. 1433–1438, 2008.

[31] M. Kuderer, S. Gulati, and W. Burgard, “Learning driving styles for autonomous vehicles from demonstration,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2641–2646.

[32] H. Gao et al., “Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making,” *Int. J. Adv. Robot. Syst.*, vol. 15, no. 6, 2018, Art. no. 1729881418817162.

[33] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, “Courteous autonomous cars,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 663–670.

[34] L. Sun, W. Zhan, and M. Tomizuka, “Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning,” in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2111–2117.

[35] Z. Huang, J. Wu, and C. Lv, “Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10239–10251, Aug. 2022.

[36] Y. Zhou, R. Fu, and C. Wang, “Learning the car-following behavior of drivers using maximum entropy deep inverse reinforcement learning,” *J. Adv. Transp.*, vol. 2020, pp. 1–13, 2020.

[37] M. Wulfmeier et al., “Large-scale cost function learning for path planning using deep inverse reinforcement learning,” *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1073–1087, 2017.

[38] Z. Zheng, “Recent developments and research needs in modeling lane changing,” *Transp. Res. Part B: Methodological*, vol. 60, pp. 16–32, 2014.

[39] S. E. Li, F. Gao, K. Li, L.-Y. Wang, K. You, and D. Cao, “Robust longitudinal control of multi-vehicle systems—A distributed h-infinity method,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2779–2788, Sep. 2018, doi: 10.1109/TITS.2017.2760910.

[40] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, “The next generation simulation program,” *Inst. Transp. Engineers J.*, vol. 74, no. 8, pp. 22–26, 2004.

[41] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, “Social behavior for autonomous vehicles,” *Proc. Nat. Acad. Sci.*, vol. 116, no. 50, pp. 24972–24978, 2019.

[42] T. Phan-Minh et al., “DriveIRL: Drive in real life with inverse reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1544–1550.

Jiangfeng Nan received the B.S. degree from North China Electric Power University, Beijing, China, in 2017, and the M.S. degree from the Beijing Institute of Technology, Beijing, China, in 2019. He is currently working toward the Ph.D. degree in vehicle engineering with Beihang University, Beijing. His research interests include intelligent driving strategy, dynamics and controls, and autonomous driving.

Weiwen Deng (Member, IEEE) received the B.S. degree from Nanjing Aeronautical Institute, Nanjing, China, in 1983, the M.S. degrees from both Beihang University, Beijing, China, and Michigan State University, East Lansing, MI, USA, in 1989 and 1998, respectively, and Ph.D. degree from Oakland University, Rochester, MI, USA, in 2004. He is currently a distinguished Professor and was the Dean of School of Transportation Science and Engineering, Beihang University. Prior to that, he was a Senior Researcher with General Motors R&D Center, USA. His primary research interests include dynamics and controls, modeling and simulation on intelligent, and electric vehicles. He is CSAE and CAAI Fellow, and the editor of serval international journals.

Ruzheng Zhang received the B.S. and Ph.D. degrees in automotive engineering and power engineering and engineering thermophysics from Tsinghua University, Beijing, China, in 2013 and 2019, respectively. Since 2019, he has been an Algorithm Researcher with Horizon Robotics, Beijing. His research interests include human-like decision making, planning, and control for automated vehicles.

Ying Wang received the B.S. and M.S. degrees from the College of Software, Jilin University, Changchun, China, in 2006 and 2008, respectively, the Ph.D. degree from the College of Computer Science and Technology, Jilin University in 2012. She is currently an Associate Professor with the College of Computer Science and Technology, Jilin University. Her research interests include autonomous driving simulation and driving environment perception.

Rui Zhao received the B.S. degree in thermal energy and dynamic engineering and the M.S. degree in power engineering from the Beijing University of Technology, Beijing, China, in 2016 and 2019, respectively. She is currently working toward the Ph.D. degree in vehicle engineering with Beihang University, Beiing. Her research interests include developing modeling steering feedback torque for steer-by wire vehicles and driving simulators, and vehicle intelligent driving strategy.

Juan Ding received the B.S. degree in physical science from Beihua University, Jilin, China, in 2005, and the M.S. and the Ph.D. degrees in condensed matter physics from Jilin University, Jilin, in 2008 and 2011, respectively. She is currently working with PanoSim Technology Company, Limited, Jiaxing, China. Her research interests include steer-by-wire vehicles and driving simulators.