Applied Intelligence (2023) 53:13192–13205
https://doi.org/10.1007/s10489-022-04172-1

Personalized lane change decision algorithm using deep reinforcement learning approach

Daofei Li[1] · Ao Liu[1]

Accepted: 12 September 2022 / Published online: 7 October 2022

Abstract

To develop driving automation technologies for humans, a human-centered methodology should be adopted for safety and satisfactory user experience. Automated lane change decision in dense highway traffic is challenging, especially when considering different driver preferences. This paper proposes a personalized lane change decision algorithm based on deep reinforcement learning. Firstly, driving experiments are carried out on a moving-base simulator. Based on the analysis of the experiment data, three personalization indicators are selected to describe the driver preferences in lane-change decisions. Then, a deep reinforcement learning (RL) approach is applied to design human-like agents for automated lane change decisions to capture the driver preferences, with refined rewards using the three personalization indicators. Finally, the trained RL agents and benchmark agents are tested in a two-lane highway driving scenario. Results show that the proposed algorithm can achieve higher consistency of lane change decision preferences than the comparison algorithm.

Keywords Reinforcement learning · Deep Q-Network · Automated driving · Lane change decision · Driver-in-the-loop experiment · Driving style

1 Introduction

In recent years, automated driving has become a hot topic in the automotive and transportation industries. Notable companies, such as Google Waymo, Baidu, and Cruise Automation, have launched their self-driving cars with SAE Level 4 automation, most of which are even for robotaxi ride-hailing services on open roads. On the other hand, the Advanced Driving Assistance Systems (ADAS) in production cars, by enhancing safety and comfort, have been widely recognized by customers, e.g., Lane Keeping Assist (LKA), and Navigate on Pilot (NOP) by NIO.

Researches on earlier-introduced ADAS functions show that personalization in driving assistance is crucial to improve both comfort and safety [1]. Before fully automated driving is possible for mass production in the future, there is no doubt that human drivers are still necessary to sit behind the wheel to supervise or even handle the daily driving tasks, though parts of which may be assisted by automation. Therefore, a well-designed driving automation strategy should be compatible with or even aligned to human preferences, to better support human drivers. Regardless of partial or full levels of driving automation, if the actual assistance in fulfilling driving tasks is congruent with personal styles in manual driving mode, the customer acceptance of assistant functions can be guaranteed.

Lane change is a common but complicated maneuver in driving, for which various driving assistance systems have been provided, such as Blind Spot Detection (for warning only), Lane Change Assist with Turn Assist, or even Auto Lane Change (ALC). Among them, ALC in particular needs to be designed in line with drivers' preferences or personal styles of maneuvering. Even in hands-free driving mode, the driver may still experience dynamic and stressful highway lane-changing scenarios on the edge of incidents or even accidents. For lane change assistance, a fundamental motivation behind ADAS personalization is that different drivers have different standards of safety perception, e.g. the acceptable gap, relative distance, and approaching rate. A universal assist design for all drivers

✉ Daofei Li
dfli@zju.edu.cn

Ao Liu
ao@zju.edu.cn

[1] Institute of Power Machinery and Vehicular Engineering, Faculty of Engineering, Zhejiang University, No 38 Zheda Road, Xihu District, Hangzhou, 310027, Zhejiang, China

may cause problems, either too conservative for aggressive drivers or the other way around [2]. For one thing, this may prevent the driver from activating the function. Moreover, if the driver cannot understand the decisions made by the ADAS function, serious accidents may happen, especially at high speeds. Therefore, it is crucial to design a safe, comfortable, and personalized algorithm for automated lane-change maneuvers.

Personalization is a feasible way to enhance drivers' trust in automation. To incorporate certain personalized styles in driving automation, a straightforward way is to imitate or even replicate a specific driver's driving operations in the addressed scenarios, that is, to model the driver behaviors. However, a generalized driver model for solving various driving tasks, though attractive, is not available yet. In consideration of the radical variation of driving conditions, driver modeling is usually done in a scenario-by-scenario fashion, i.e., a holistic driver model is an integration of several sub-models of driver behaviors, e.g., car following, lane change, steering, etc. [3].

For lane change decision modeling, there are two popular methods, mechanism-based and learning-based models [4–6]. As for mechanism-based models, Gipps' model [7] summarized the lane change decision-making process as a flowchart, in which any factors that affect lane change decision-making can be added or replaced. Although there is no consideration of drivers' variational behavior, it still has a profound influence on the subsequent research. In [8], lane change behaviors are classified into three categories. Although several factors, such as motivation, advantage, and urgency, are considered, the final execution still depends on the availability of the gap between the preceding and the following vehicles in the target lane. Models applied in the FRESIM [9] and NETSIM [10] are similar but only different in the ways of calculating the acceptable gap. Kesting [11] proposed a general lane change decision-making model, MOBIL, in which the IDM model [12] is used to compare the total deceleration of all surrounding vehicles before and after lane change, and then the decision is made. A parameter called "policy factor" is considered in MOBIL to reflect the cooperation between drivers. Additionally, in the case of merging and congested scenarios, game theory is used to model the lane change decision considering the interaction with other vehicles [13, 14]. However, these approaches may not be easily applied to scenarios with several surrounding cars.

Recently, many learning-based approaches have been proposed for unmanned vehicles, e.g., road vehicles [15], surface vehicles [16], and aerial vehicles [17, 18]. In terms of the lane change decision modeling of automated driving, Vallon et al. [19] propose a data-driven approach to capture the lane change decision behavior of the human driver with the help of Support Vector Machine (SVM) classifiers. The results show that the personalized algorithm can reproduce the behaviors of different drivers without explicit initiation. Mirchevska et al. [20] design an RL agent using a Deep Q-Network, which can drive as closely as possible to the desired velocity. Hoel et al. [21] train a Deep Q-Network agent for a truck-trailer combination in highway driving scenarios, which can finish overtaking maneuvers better than the commonly-used reference model consisting of IDM and MOBIL. Learning-based approaches can include more influential factors in lane change decision modeling, but to consider the driver preferences, a lot more lane change data for a particular driver is required. These approaches are inherently data-hungry, while the data size, coverage, and details, will determine the model scope and the potential application domains. There are already some public datasets of driving study, e.g., the Next Generation SIMulation (NGSIM) program launched by the Federal Highway Administration (FHWA) [22] and the highD dataset published by RWTH Aachen University [23]. Regardless of whether the data collection is via cameras mounted in hovering drones or cameras fixed on traffic sign poles, for a specifically investigated vehicle-driver combination, only a very limited range of driving trajectories are available. Based on these data, most previous researches only consider the basic problem in lane change decision modeling, i.e. the general behaviors for safety and efficiency. Due to the lack of data at the driver-specific level, the personalized preferences of in-vehicle drivers have not been well studied.

To overcome these limits, this work focuses on personalized automated lane change decision-making in a two-lane highway scenario. Figure 1 shows how the personalized decision algorithm is achieved from raw data collection, analysis, reinforcement learning (RL) algorithm design, to validation. More specifically, the RL agents are designed to make lane change decisions based on environmental perception and the personalized reward function. The main contributions of this paper are two-fold.

(1) By analyzing the simulator driving data, three effective indicators of driver lane change decision preferences are determined, i.e., time to collision with the front car in the current lane (t_f), time to collision with the front car in the target lane (t_{nf}), and the relative speed with the rear car in the target lane (dv_{nb}).

(2) Based on the three indicators of driver preferences, a personalized decision-making algorithm is proposed using Deep Q-Network. The comparative results show that the proposed algorithm can perform better than the benchmark algorithm with a commonly-used policy.

The rest of the paper is organized as follows. Section 2 formulates the lane change decision problem and introduces the basics of reinforcement learning. Section 3 presents

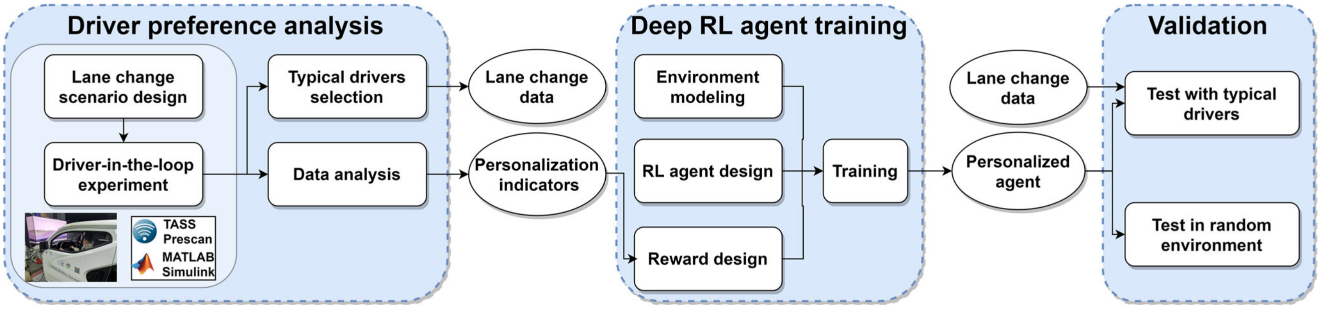

Driver preference analysis
Lane change scenario design
Driver-in-the-loop experiment
TASS Prescan
MATLAB Simulink
Typical drivers selection
Data analysis
Lane change data
Personalization indicators
Deep RL agent training
Environment modeling
RL agent design
Reward design
Training
Lane change data
Personalized agent
Validation
Test with typical drivers
Test in random environment

Fig. 1 The proposed algorithm structure of personalized lane change decision

the driver-in-the-loop experiments and driver preference analysis. The RL-based lane change decision design is detailed in Section 4, while the results are summarized in Section 5. Finally, the conclusion and some potential future work are given in Section 6.

2 RL formulation of decision problem

For ADAS or automated driving, it is a complex task to make a lane change decision with multiple surrounding cars, especially considering driving personalization. As shown in Fig. 1, there are three steps to tackle the problem. Step 1 is to obtain the human driving data and personalization indicators through driver-in-the-loop experiments. Steps 2 and 3 are to design and validate the personalized RL agents. Here, the various personalization indicators need to be considered simultaneously, which can reflect different decision preferences by different trade-off strategies. On the other hand, the algorithm should have adaptability in practical applications due to environmental conditions and driver data. Therefore, the RL approach is adopted to model the driver preferences in lane change decisions.

2.1 Lane change decision problem

In this RL problem of lane change decision task, Deep-Q-Network is used to learn the state-action value function in describing the personalized reward, $Q(s, a)$. As a typical example, a two-lane highway scenario with three surrounding cars at random speeds is considered, as shown in Fig. 2. The notations of surrounding cars are as follow: the ego car (C_{ego}), the front car in the current lane (C_f), the front car in the target lane (C_{nf}), and the rear car in the target lane (C_{nb}). Considering the influences of the surrounding cars, the ego car's task is to decide whether or when to change from the current lane to the target lane.

2.2 Reinforcement learning

In an RL problem, an agent selects an action a depending on the current state of the environment s and the policy π, then the environment will change to a new state s' and return a reward r. The goal is to find an optimal policy π^* that maximizes the cumulative reward R_t, defined as

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \tag{1}$$

where r_{t+k} is the reward returned at step $t + k$, and γ is a discount factor, $\gamma \in [0, 1]$.

State-action value function $Q(s, a)$ is used to evaluate the expected cumulative reward of agent when selecting action a in state s, that is

$$Q(s, a) = E[R_t | s_t = s, a_t = a]. \tag{2}$$

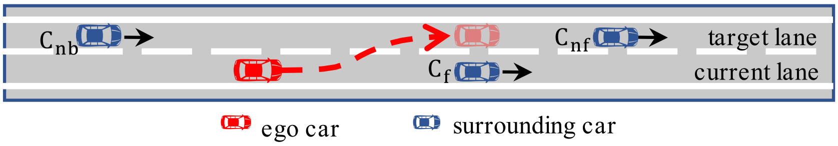
Q-learning is a classical algorithm for the problem with limited states and actions, the state-action values are saved in a Q-table. The optimal state-action value function in Q-learning is

$$Q^*(s, a) = E[r + \gamma \max_{a'} Q^*(s', a') | s_t = s, a_t = a]. \tag{3}$$

When $Q^*(s', a')$ is known, the optimal policy is to select an action a' that maximizes $Q^*(s', a')$.

However, if the state space is continuous, it is impractical to remember all the state-action values with a table. To handle this, the Deep Q-Network (DQN) [24] is adopted, which can approximate the optimal state-action value $Q^*(s, a)$ with a nonlinear estimator $Q(s, a; \theta)$. Network weights θ will be updated during the training process to

Fig. 2 Lane change decision scheme in a two-lane driving scenario



minimize the following loss function

$$L(\theta) = E[(r + \gamma \max_{a'} Q(s', a', \theta) - Q(s, a, \theta))^2]. \tag{4}$$

To avoid unstable training caused by the same network weight, the weight of target network is set to θ^- and replaced by the prediction network weights θ at every fixed-step. Then the final loss function is defined as

$$L(\theta) = E[(r + \gamma \max_{a'} Q(s', a', \theta^-) - Q(s, a, \theta))^2]. \tag{5}$$

3 Driver preferences analysis

To model driver preferences, an ideal dataset should be from several different drivers' naturalistic driving on public roads. However, considering lane change as one kind of highly dynamic and time-critical process, if the lane change decision timing matters, it is difficult to record the exact decision processes. As one improved way of data collection, experimental driving on test roads may better assure the realistic driving conditions and the drivers can almost maneuver the vehicle as they usually do, but only if the surrounding cars (usually 2 to 3 additional cars) can be coordinated well by human or robot drivers. However, experimental driving, if with only limited time and/or limited financial budget, is still impossible to collect enough data on lane change decision strategies, not to mention that the experiment safety risk is extremely high due to the involvement of multiple vehicles at high speeds. Therefore, to analyze the driver preferences in lane change decisions, a moving-base driving simulator is adopted for the original data collection.

3.1 Driver-in-the-loop experiments

A driver-in-the-loop (DIL) experiment environment is designed based on a 6-Degrees-of-Freedom (6DoF) driving simulator, which can provide a realistic driving experience, as shown in Fig. 3. The two-lane highway driving scenario is constructed in the simulation software, TASS Prescan, while the surrounding cars with random constant speeds are controlled by the MATLAB/Simulink model. The ego car is controlled by the human driver with a real steering wheel and gas/brake pedal. A specific button on the steering wheel is set to record the timestamp of every lane change initiation, while the other recorded data include the position and speed of all vehicles during the whole process.

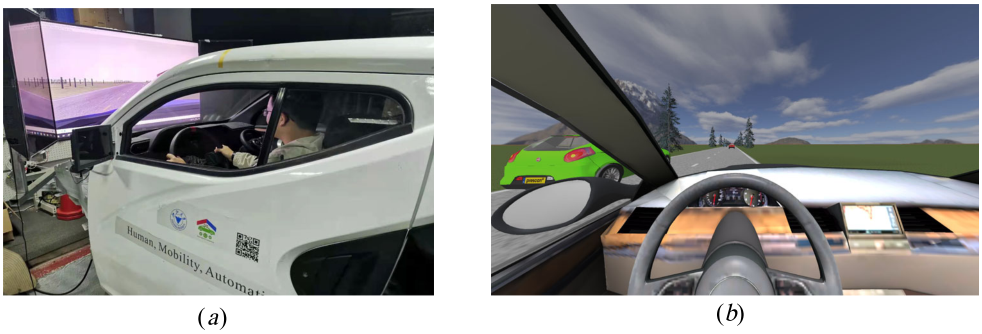
Ten drivers, aged between 20 and 26, are invited to participate in the DIL experiments. Each driver is asked to make the lane change maneuvers 50 times on four sections with different speed limits, i.e., 60kph, 70kph, 80kph, and 90kph. Since not all drivers can exactly follow the speed limits, a maximum 5kph speed error over limits is still considered acceptable.

According to the existing research on natural driving data [25, 26], there are many factors affecting drivers' lane change decisions, such as relative velocity, time to collision (TTC), and relative distance. For safe lane change, a driver should try to avoid collisions with surrounding cars. For the front cars in the current and target lanes, drivers mainly judge whether a collision will occur by sensing the relative speed and distance, which can be described by the value of TTCs, t_f, and t_{nf}, respectively. As for the rear car in the target lane, the approaching rate is adopted as a judgment indicator, which is included in the form of relative speed, dv_{nb}. Therefore, a different driver style in lane change decision corresponds to a different combination of values of these three personalization indicators, i.e., t_f, t_{nf} and dv_{nb}.

Based on the simulator driving data, Fig. 4 shows the statistical results of three indicators in different speed ranges. It is found that these three indicators are positively correlated with the velocity of ego car, v_e. This phenomenon is understandable due to driver's risk perception at different speeds. Further, correlation analysis is used to check these indicators' relationships, while the detailed analysis results for each specific driver are given in Table 1. The results reveal that there is a linear correlation between v_e and three indicators ($p < 0.05$) in 80% of the drivers. Therefore, the driver personalization in lane change decisions can be defined using these three indicators. Then the personalization indicator set is

$$I_{dp} = [t_f, t_{nf}, dv_{nb}]^T. \tag{6}$$

Fig. 3 (a) The 6-DoF driving simulator. (b) Lane change scenario in Prescan



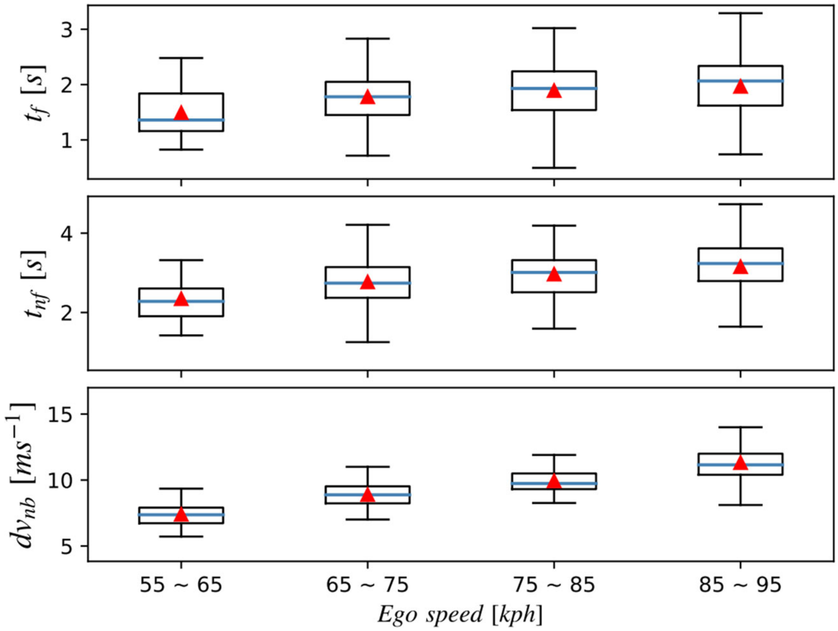


Fig. 4 The statistical results of lane change decision data in DIL experiment in different speed ranges. (blue lines represent median values of indicators in different speed ranges; red triangles represent mean values.)

These decision points are fitted with linear regression as

$$I_{dp} = Av_e + b, \tag{7}$$

which will be used as reference lines in the reward function design of Section 4.

To justify the effectiveness and rationality of (7), a naturalistic driving dataset, HighD, is adopted to illustrate the universality of the experimental samples. The HighD is the abbreviation of the Highway Drone dataset, which is a large-scale naturalistic vehicle trajectories dataset recorded at German highways and covers over 110 thousand vehicles, 44.5 driven kilometers, and 147 driven hours [23]. A total of 1,469 lane change maneuvers are extracted, of which 297 cases that match the scenario in Fig. 2 are finally selected. The scatter diagram and statistical summary of ego speed and three personalization indicators are given in Fig. 5, in which the correlation analyses are also indicated, respectively. Similar to Fig. 4, the linear correlation between indicators and ego speed, with all significance levels $p < 0.05$, justifies that (7) also holds true in naturalistic driving cases.

3.2 Drivers with typical preferences

Based on the DIL experiments data, the driver lane change decisions are clustered into three groups, i.e., Defensive, Normal, and Aggressive, as shown in Fig. 6. When initiating a lane change maneuver, a more aggressive driver corresponds to the smaller TTCs (t_f, t_{nf}) and relative speed(dv_{nb}).

According to the clustering results, three drivers with obvious differences are selected as typical examples, as shown in Fig. 7. For each driver, the indicators, t_f, t_{nf} and dv_{nb} increase linearly with the speed of ego car, v_e. Then,

Table 1 Correlation analysis for all drivers (t_f / t_{nf} / dv_{nb})

Driver	Coefficient of determination(R^2)	F-value	p-value
0	0.126 / 0.220 / 0.789	6.898 / 13.528 / 179.937	0.0115 / 0.0006 / 0.0000
1	0.005 / 0.132 / 0.867	0.176 / 5.757 / 247.336	0.6769 / 0.0214 / 0.0000
2	0.594 / 0.621 / 0.806	55.688 / 62.226 / 157.907	0.0000 / 0.0000 / 0.0000
3	0.510 / 0.584 / 0.807	49.960 / 67.304 / 200.265	0.0000 / 0.0000 / 0.0000
4	0.072 / 0.068 / 0.734	4.132 / 3.847 / 145.940	0.0471 / 0.0551 / 0.0000
5	0.157 / 0.170 / 0.756	7.097 / 7.802 / 118.038	0.0113 / 0.0081 / 0.0000
6	0.298 / 0.383 / 0.692	19.496 / 28.581 / 103.316	0.0000 / 0.0000 / 0.0000
7	0.097 / 0.124 / 0.591	3.338 / 4.371 / 44.740	0.0773 / 0.0449 / 0.0000
8	0.084 / 0.089 / 0.822	4.383 / 4.682 / 222.194	0.0416 / 0.0355 / 0.0000
9	0.004 / 0.084 / 0.574	0.153 / 3.469 / 51.286	0.6978 / 0.0703 / 0.0000

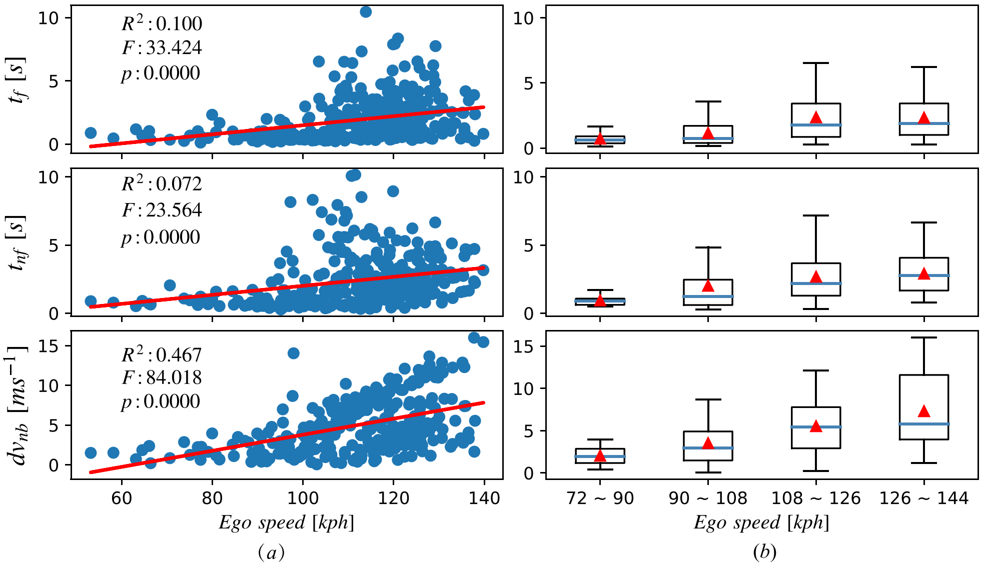


Fig. 5 The statistical results of lane change data extracted from HighD dataset. (a) The scattered points and regression curves via ordinary least square (OLS). (b) The statistical results in different speed ranges. (blue lines represent median values of indicators in different speed ranges; red triangles represent mean values.)

the curve parameters, A and b are obtained via the ordinary least square (OLS) method and presented in Table 2.

4 RL-based lane change decison

Figure 8 schemes the RL-based decision process. The action space, state space, reward function, and the Deep Q-Network are defined first. The RL environment provides information about the ego car and the surrounding cars to the decision module, in which the information is transitioned to the state vector as an input of the Deep Q-Network. Then the network outputs the state-action value for each action, and the best action is selected as the final decision.

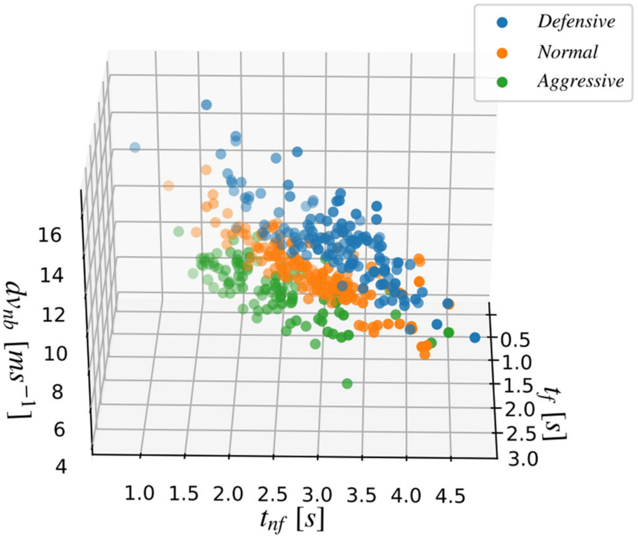


Fig. 6 The clustering result of three driver styles

4.1 Action space

Here, the left or right lane change decisions are treated as the same. There are two discrete actions in the lane change decision-making problem, i.e., a_1:TO CHANGE to the target lane, and a_2: NOT TO CHANGE lane (to stay in the current lane). Then, the action space A is defined as

$$A = \{a_1, a_2\}. \tag{8}$$

4.2 State space

Based on the analysis in Section 3, the personalization indicators, i.e., t_f, t_{nf} and dv_{nb}, need to be obtained from the state of environment. To facilitate real car applications, the variables in state space should be directly measurable via the onboard sensors. The state consists of two parts, the ego car information and the surrounding car information. Each car's information includes its longitudinal velocity v and longitudinal position x. For better performances in training, v and x are normalized to (0, 1], respectively. Therefore, the state can be described as a vector of eight normalized values,

$$s = [v_e, x_e, v_f, x_f, v_{nf}, x_{nf}, v_{nb}, x_{nb}]. \tag{9}$$

4.3 Reward functions

To train the human-like RL agents, the personalization indicators are used as the reference to design the reward functions. To help the agent trade-off the benefits between different decisions and learn to make a better choice, the sum of two actions' rewards is kept as a constant at every decision step. Therefore, the reward function for each indicator is designed as follows. If the decision is TO

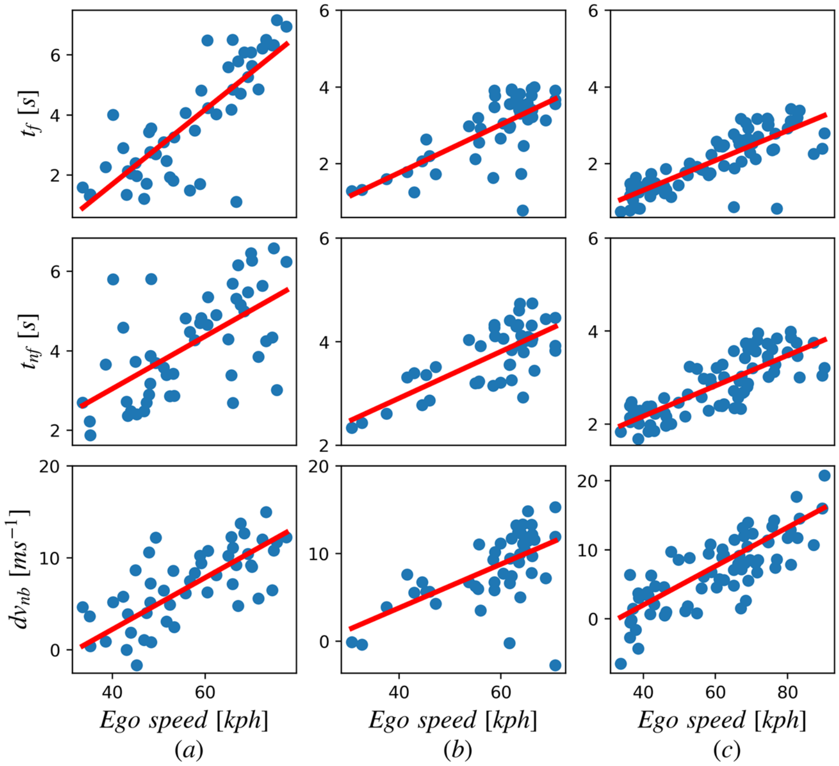


Fig. 7 The relations of three personalization indicators with ego speed at the lane change decision point. The red lines are the regression curves obtained by OLS. (a) Defensive, (b) Normal, (c) Aggressive

CHANGE lane, i.e. $action = a_1$,

$$r = \begin{cases} 1 & , e_i \in [0, m] \\ \frac{1}{m-n} * e_i - \frac{n}{m-n}, & e_i \in (m, n) \\ 0 & , e_i \in [n, +\infty) \end{cases} \tag{10}$$

If the decision is NOT TO CHANGE lane, i.e. $action = a_2$,

$$r = \begin{cases} 1 & , e_i \in [0, m] \\ \frac{1}{n-m} * e_i - \frac{m}{n-m}, & e_i \in (m, n) \\ 0 & , e_i \in [n, +\infty) \end{cases} \tag{11}$$

For each indicator $i \in I_{dp}$ in (6), e_i is the absolute error of actual value i_{act}, and reference value i_{ref},

$$e_i = |i_{act} - i_{ref}|. \tag{12}$$

The smaller e_i means the indicator i is more suitable for a personalized lane change. If the agent chooses to change lanes with a smaller e_i, it will receive a greater reward. Instead, if e_i is large enough, the agent needs to keep in the current lane for a greater reward.

m and n are two preset parameters, which represent the maximum acceptable error and the maximum effective error, respectively. If $e_i \leq m$, it means the current value of this indicator is exactly matched with the driver, while $e_i \geq n$ means this indicator does not match the driver at all. However, the extreme pursuit of all indicators will lead to the non-convergence of training and get an unsatisfactory result eventually, so, it is necessary to choose the appropriate m and n. Here, considering the range and precision of each indicator, we have $m = 0.2, n = 2$ for t_f, t_{nf} and $m = 0.5, n = 5$ for dv_{nb}.

Finally, the reward functions for all three indicators, r_f, r_{nf} and r_{nb}, are obtained and the total reward function R is defined as

$$R = r_f + r_{nf} + r_{nb}. \tag{13}$$

4.4 Neural network design and training details

Convolutional neural networks (CNN) are usually used in the architecture design with image matrix inputs. Here, the network input is a vector consisting of a series of vehicle states, i.e., the state space. Therefore, a fully connected neural network (FCNN) architecture [27] is designed for

Table 2 Parameter matrices for different driver styles via OLS

Style	A	b
Defensive	$[0.446, 0.237, 1.012]^T$	$[-3.258, 0.418, -9.015]^T$
Normal	$[0.226, 0.162, 0.898]^T$	$[-0.753, 1.105, -6.180]^T$
Aggressive	$[0.140, 0.118, 1.012]^T$	$[-0.249, 0.858, -9.251]^T$

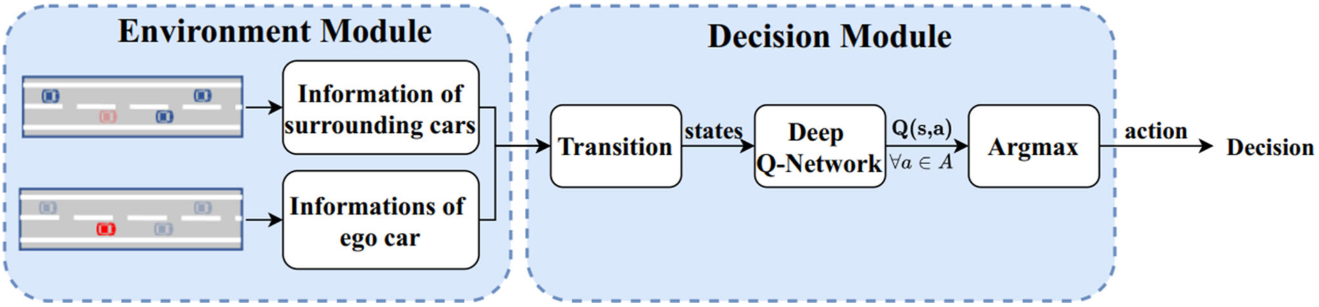


Fig. 8 The process of RL lane change decision making

the target network and the prediction network mentioned in Section 2. As shown in Fig. 9, there are three hidden layers, while each layer has 128 neurons, and the activation function of rectified linear units (ReLUs) is used. The input is a state vector of 8 × 1 size, and the output is a state-action value vector of 2 × 1 size. At time t, the neural network gets an environment state input s_t and outputs the estimation state-action values $Q(s, a)$ for each action a_i in action space A.

The neural network is trained with a learning rate η by using the DQN algorithm. The ϵ-greedy policy is applied in training, and along with the training process, the value of ϵ will decrease from ϵ_s to ϵ_e linearly. The factor γ is used to consider the discount of future rewards. The replay memory with a size of M_r is set. The training begins after the initial minimal memory size M_i, and the random sample mini-batch size is set to M_m. The weights of the target network θ^- are replaced by the prediction network's weights θ every N_u episode, and they are both initialized with a standard normal distribution denoted as $N(0, 1)$. Every episode starts with a random environment state and stops with a TO CHANGE lane decision or maximal episode step N_s. The maximal training episode is set to N_e.

Based on the commonly-used setting in DQN, here the hyper parameters are adjusted and determined according to our pre-defined lane change environment. Specifically, the sample time in the RL environment is 0.05s and the agent needs to finish the lane change maneuver in 10s, so the maximal episode step (N_s) should be 200. The maximal training episode (N_e) is set to 10,000, which is large enough to allow the full convergence of training. The initial minimal memory size (M_i) and the replay memory size (M_r) are set relatively small because of the action space design and the short episode step. As for the discount factor (γ), it can be determined with $\gamma \approx 0.01^{\frac{1}{200}}$, which means the 200th step reward accounts for 0.01 of the total reward. And the rest hyper parameters, i.e., learning rate (η), initial exploration (ϵ_s), final exploration (ϵ_e) and target network update frequency (N_u), are determined by the grid search method. The hyper parameters used in training are summarized in Table 3.

5 Results and discussion

In this paper, three kinds of personalized RL agents are designed and trained for lane change decision-making to reproduce the typical drivers' preferences. In the lane change scenario, agents experience different states and learn to make decisions themselves by repeating the lane change

Fig. 9 The designed FCNN architecture

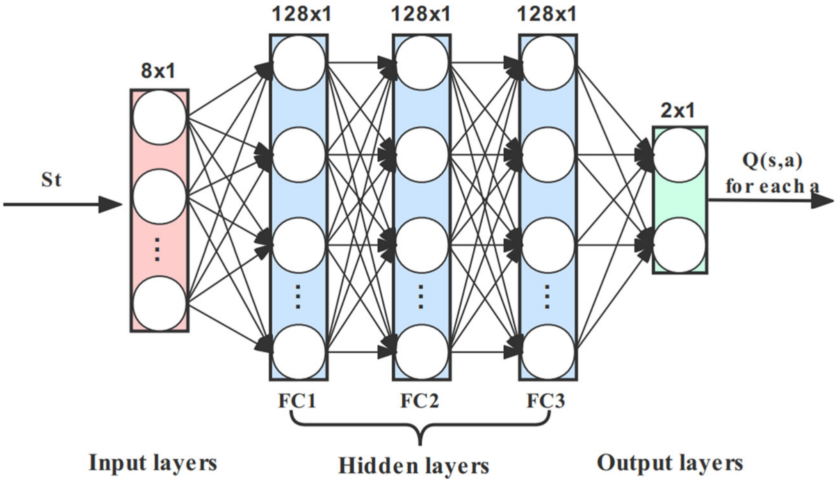


Table 3 Hyper parameters setting

Symbol	Parameters	Value
η	Learning rate	0.005
ϵ_s	Initial exploration	0.8
ϵ_e	Final exploration	0.1
γ	Discount factor	0.98
M_r	Replay memory size	10000
M_i	Initial minimal memory size	2000
M_m	Mini-batch size	32
N_u	Target network update frequency	20
N_s	Maximal episode step	200
N_e	Maximal training episode	10000

interactions. Finally, a stable policy for any state can be learned.

5.1 Training results

Figure 10 shows the training results of three RL agents with different lane change decision preferences. The horizontal axis is the training episode, the training losses defined in (5) are in the left column, and the step rewards defined in (13) are in the right column. It is obvious that the loss curve has a quick downtrend in the first 1,500 training episodes and then flattens out, which means the neural network converges. The average step reward is increased in every episode during the training process, implying that the RL agents have learned to select the actions with a higher reward in a series of lane change maneuvers. Both the loss and step reward stabilize eventually. Due to the random training environment, the curve of the step reward is not very smooth, but its mean value reaches stable at around 1.4.

5.2 Benchmark algorithm

This paper focuses on the personalization decisions, and the ability of RL agents to make optimal decisions during lane change needs to be proved, especially when considering three indicators at the same time. Therefore, a benchmark algorithm is set, to show the advantage of the proposed algorithm. With the designed reward function and the selected typical drivers, the benchmark algorithm directly compares the rewards of two actions at every step, and then makes the decision with a higher reward. This is a kind of greedy strategy commonly used, and it will be deployed on three benchmark agents with Defensive, Normal and Aggressive styles. Further, they will be tested in the same simulation environment together with the trained RL agents.

5.3 Test and validation

The trained RL and benchmark agents are tested in a random simulation environment. They make lane change decisions considering the environment states, and then the values of personalization indicators are recorded. Then three sets of lane-change points are obtained. The reference lines and the lane-change points are compared, then the similarities are calculated by the Mean Absolute Error (MAE). To further illustrate the agent's personalized preferences, the statistical results of decision-making accuracy are presented through the comparisons among the typical drivers, the trained RL agents, and the benchmark agents.

Figure 11 shows the test results of three personalized RL agents and benchmark agents compared to the typical drivers with different driving styles, i.e., Defensive, Normal, and Aggressive. When the agents decide to make a lane change, the points in sub-plots represent the value of personalization indicators at different speeds of the ego car, while the blue and orange points are generated by RL agents and benchmark agents, respectively.

To describe how closely the agents and the corresponding drivers decide in lane change maneuvers, the similarities between reference lines and lane-change points are represented by the MAE, which is defined as

$$MAE = \frac{1}{n}\sum_{i=0}^{n} |y_{a_i} - y_{r_i}|, \tag{14}$$

where y_{a_i} is the actual value of indicators, and y_{r_i} is the reference value obtained from the reference line with the same speed as y_{a_i}. As can be seen in Fig. 11, for all three personalized RL agents, their lane-change points are close to the reference lines. As for the benchmark agents, only the values of dv_{nb} are close to the references, while the

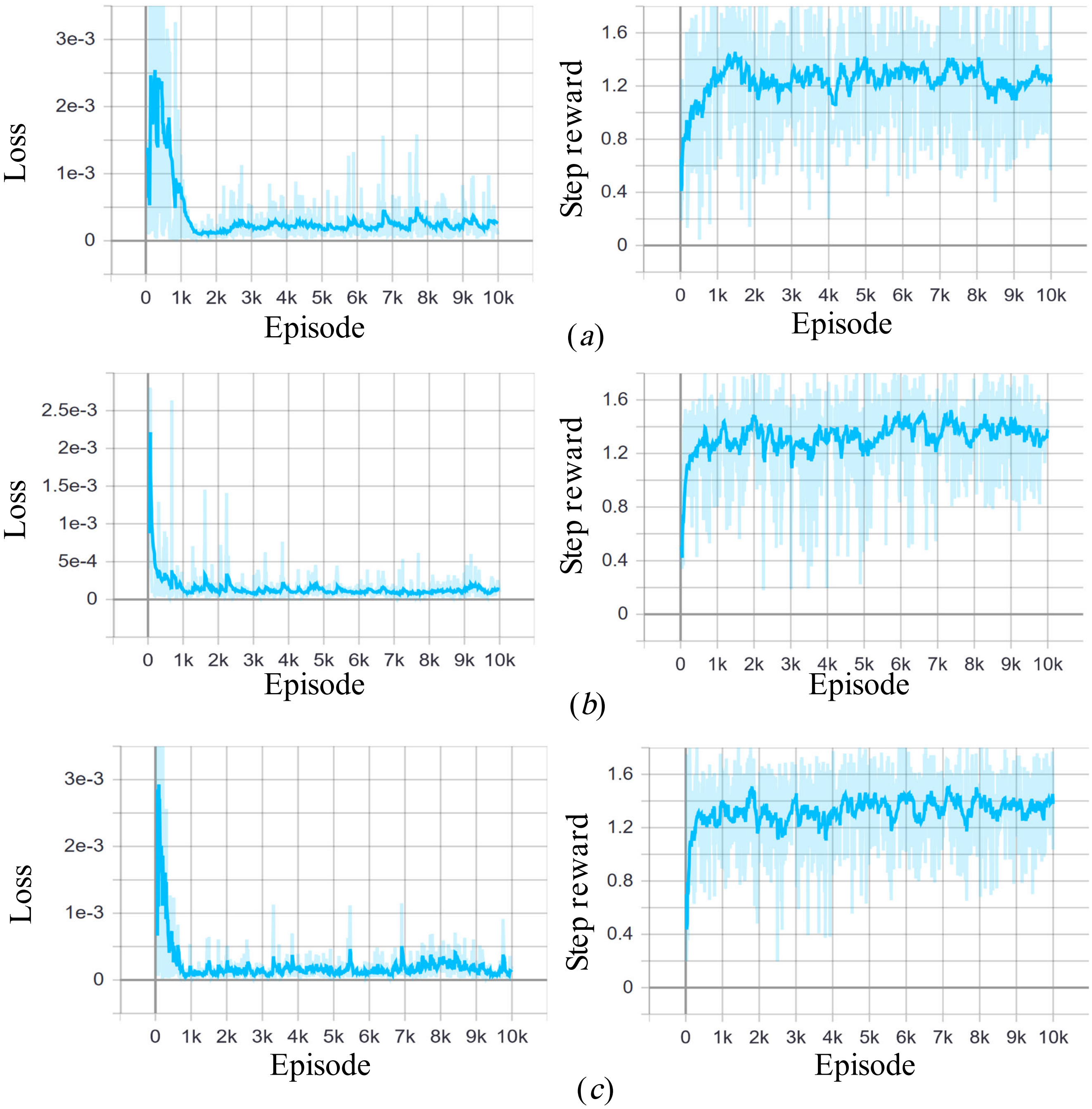


Fig. 10 Training losses (left column) and step rewards (right column) for personalized RL agents. The dark blue curves are obtained by smoothing the real values in light blue color. (a) Defensive, (b) Normal agent, (c) Aggressive

performance of the other two indicators, t_f, t_{nf}, are worse than that of the RL agents. Specifically, at the initiation of lane change decision, the RL agents' MAEs of t_f, t_{nf} are obviously less than that of the benchmark agents.

Furthermore, the decision results of typical drivers, trained RL agents, and benchmark agents in a series of the same states are compared, with the statistical results shown in Fig. 12. The blue circle points represent that the human drivers and the agents make the same decisions, while the red triangles, in contrast, represent their opposite decisions. For RL agents, we have 95.9% accuracy for Defensive agent, 100% accuracy for Normal agent, and

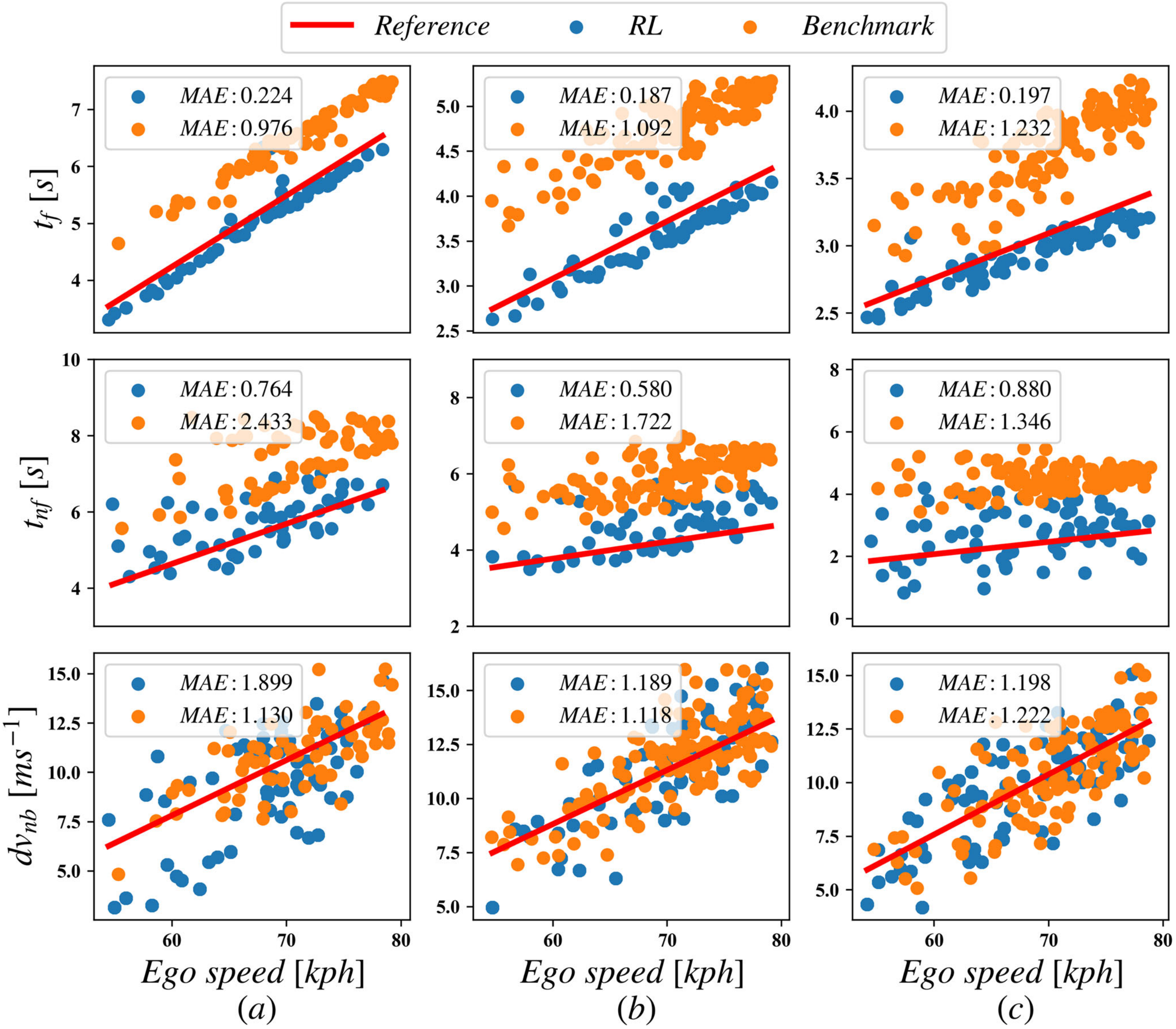


Fig. 11 Test results of personalization indicators in lane change decision at different ego speeds. (a) Defensive agents. (b) Normal agents. (c) Aggressive agents

98.6% accuracy for Aggressive agent. In contrast, the values of accuracy for benchmark agents are only 83.7%, 87%, and 86.5%, respectively. For Defensive and Aggressive RL agents, there are totally three opposite cases marked in Fig. 11, compared with original lane change data generated by typical drivers. To be specific, in cases 1 and 2, the relative distance (d_{nb}) between the ego car (C_{ego}) and the car behind in the target lane (C_{nb}) is too large, 142.36 meters and 126.11 meters, respectively. However, in this training environment, if V_{nb} is too far away from the $C_{ego}(d_{nb} > 100)$, it can be considered that C_{nb} has no effect on the lane change decision making of C_{ego}. As for case 3, there is no leading car (C_{nf}) in the target lane (the related indicator t_{nf} is recorded as -1), which contributes to a false result of the decision. All these failed cases' environment states are not involved in the training environment, so the agent cannot handle them.

To sum up, with the similarity comparisons, the decision accuracies, and the failed case analysis, it is proved that the personalized RL agents can make the lane change decision

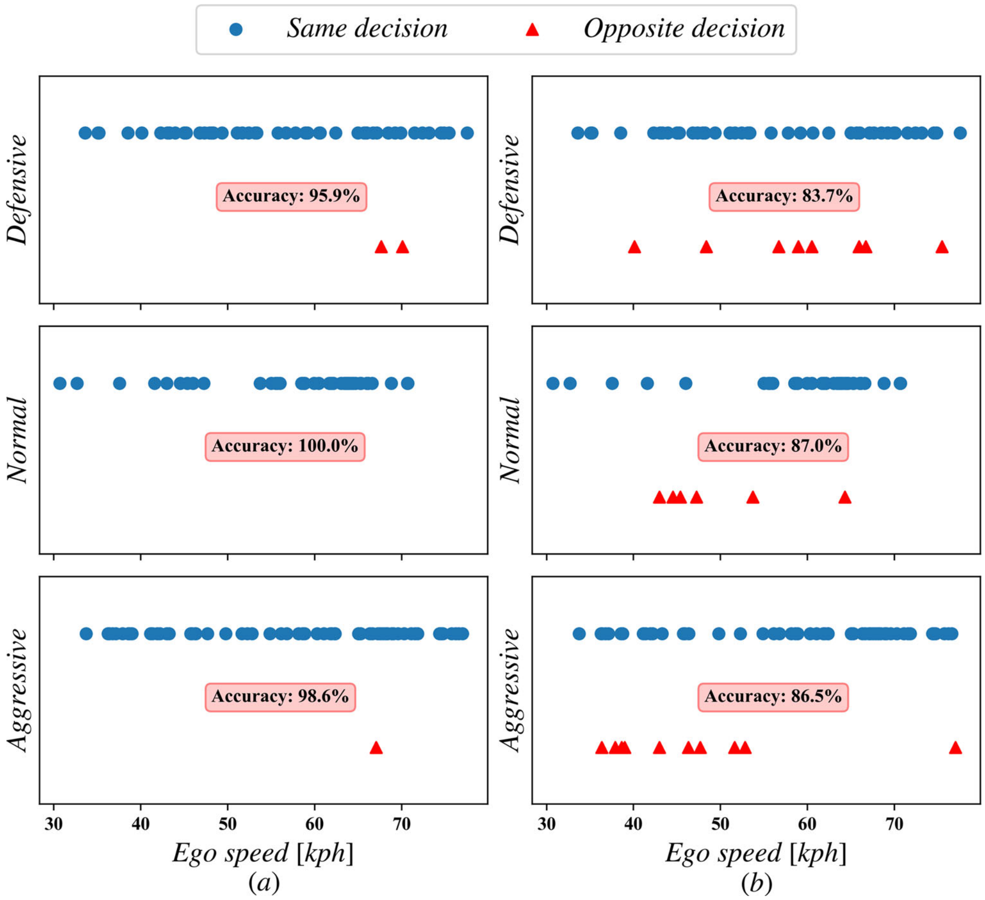


Fig. 12 The comparisons of lane change decision making results between human drivers, trained RL agents and benchmark agents with three different personalized preferences. (a) RL agents. (b) Benchmark agents

more like human drivers with three different styles than the benchmark agents.

6 Conclusion

Due to the stressful dynamics, lane change is a common but difficult decision task in dense traffic, especially in highway scenarios. For better user experience, automated driving requires further consideration of driver personalized preferences.

This paper proposes a personalized decision algorithm for lane change based on RL. The RL agent can successfully reproduce the driver's preferences for lane change decisions, which is promising for further applications in the human-centered design of automated driving.

This work is a part of ongoing research on personalized driving automation considering user experience. There are some limits to overcome in future studies. For example, only three personalization indicators are selected, while there may exist some other indicators that affect lane change decisions, e.g., more variables of the traffic and road conditions. For applications, the algorithm may be further extended by enriching the state space design with the driver's high-level preferences according to the driver's current status, e.g., based on driver state monitoring systems.

Funding The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Department of Science and Technology of Zhejiang (No. 2022C01241, 2018C01058).

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interests The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Hasenjager M, Heckmann M, Wersing H (2020) A survey of personalization for advanced driver assistance systems. IEEE Trans Intell Veh 5(2):335–344. https://doi.org/10.1109/TIV.2019.2955910
2. Butakov V, Ioannou P (2015) Personalized driver/vehicle lane change models for ADAS. IEEE Trans Veh Technol 64(10):4422–4431. https://doi.org/10.1109/TVT.2014.2369522
3. Macadam CC (2003) Understanding and modeling the human driver. Veh Syst Dyn 40(1-3):101–134. https://doi.org/10.1076/vesd.40.1.101.15875
4. Moridpour S, Sarvi M, Rose G (2010) Lane changing models: a critical review. Transp Lett 2(3):157–173. https://doi.org/10.3328/TL.2010.02.03.157-173
5. Rahman M, Chowdhury M, Xie Y, He Y (2013) Review of microscopic lane-changing models and future research opportunities. IEEE Trans Intell Transp Syst 14(4):1942–1956. https://doi.org/10.1109/TITS.2013.2272074
6. Toledo T (2007) Driving behaviour: models and challenges. Transp Rev 27(1):65–84. https://doi.org/10.1080/01441640600823940
7. Gipps PG (1986) A model for the structure of lane-changing decisions. Transp Res B: Methodol 20(5):403–414. https://doi.org/10.1016/0191-2615(86)90012-3
8. Halati A, Lieu H, Walker S (1997) CORSIM-Corridor traffic simulation model. In: Traffic congestion and traffic safety in the 21st century, chicago, illinois, pp 570–576
9. Holm P, Tomich D, Sloboden J, Lowrance C (2007) Traffic analysis toolbox vol IV: guidelines for applying CORSIM microsimulation modeling software traffic simulation
10. Belitsky V, Krug J, Neves EJ, Schütz GM (2001) A cellular automaton model for two-lane traffic. J Stat Phys 103(5):945–971. https://doi.org/10.1023/A:1010361022379
11. Kesting A, Treiber M, Helbing D (2007) General lane-changing model MOBIL for car-following models. Transp Res Rec: J Transp Res Board 1999(1):86–94. https://doi.org/10.3141/1999-10
12. Treiber M, Hennecke A, Helbing D (2000) Congested traffic states in empirical observations and microscopic simulations. Phys Rev E 62(2):1805–1824. https://doi.org/10.1103/PhysRevE.62.1805
13. Kita H (1999) A merging–giveway interaction model of cars in a merging section: A game theoretic analysis. Transp Res A Policy Pract 33(3-4):305–312. https://doi.org/10.1016/S0965-8564(98)00039-1
14. Pei Y, Xu H (2006) The control mechanism of lane changing in Jam condition. In: 2006 6th World congress on intelligent control and automation, pp 8655–8658. IEEE. https://doi.org/10.1109/WCICA.2006.1713670. http://ieeexplore.ieee.org/document/1713670/
15. Li G, Li S, Li S, Qin Y, Cao D, Qu X, Cheng B (2020) Deep reinforcement learning enabled decision-making for autonomous driving at intersections. Auto Innovation 3(4):374–385. https://doi.org/10.1007/s42154-020-00113-1
16. Song D, Gan W, Yao P, Zang W, Zhang Z, Qu X (2022) Guidance and control of autonomous surface underwater vehicles for target tracking in ocean environment by deep reinforcement learning. Ocean Eng 250:110947. https://doi.org/10.1016/j.oceaneng.2022.110947
17. Qu C, Gai W, Zhong M, Zhang J (2020) A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. Appl Soft Comput 89:106099. https://doi.org/10.1016/j.asoc.2020.106099
18. Eroglu B, Sahin MC, Ure NK (2020) Autolanding control system design with deep learning based fault estimation. Aerosp Sci Technol 102:105855. https://doi.org/10.1016/j.ast.2020.105855
19. Vallon C, Ercan Z, Carvalho A, Borrelli F (2017) A machine learning approach for personalized autonomous lane change initiation and control. In: 2017 IEEE Intelligent vehicles symposium (IV), pp 1590–1595. IEEE. https://doi.org/10.1109/IVS.2017.7995936. http://ieeexplore.ieee.org/document/7995936/
20. Mirchevska B, Pek C, Werling M, Althoff M, Boedecker J (2018) High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp 2156–2162. IEEE. https://doi.org/10.1109/ITSC.2018.8569448 . https://ieeexplore.ieee.org/document/8569448/. Accessed 25 June 2021
21. Hoel C-J, Wolff K, Laine L (2018) Automated speed and lane change decision making using deep reinforcement learning. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp 2148–2155, IEEE. https://doi.org/10.1109/ITSC.2018.8569568. https://ieeexplore.ieee.org/document/8569568/. Accessed 15 Dec 2021
22. Alexiadis V, Colyar J, Halkias J, Hranac R, Mchale G (2004) The next generation simulation program. ITE Journal 74(8):22–26. https://doi.org/10.1111/j.1399-0039.2009.01336.x
23. Krajewski R, Bock J, Kloeker L, Eckstein L (2018) The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp 2118–2125. IEEE. https://doi.org/10.1109/ITSC.2018.8569552. https://ieeexplore.ieee.org/document/8569552/. Accessed 19 Nov 2020
24. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533. https://doi.org/10.1038/nature14236
25. Olsen ECB (2003) Modeling Slow Lead Vehicle Lane Changing. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia. https://vtechworks.lib.vt.edu/handle/10919/29889. Accessed 28 June 2021
26. Bogard SE (1999) Analysis of data on speed-change and lane-change behavior in manual and ACC driving. Technical report, U.S. Department of Transportation National Highway Traffic Safety Administration, Washington, DC, USA. http://www.researchgate.net/publication/30818139_Analysis_of_data_on_speed-change_and_lane-change_behavior_in_manual_and_ACC_driving. Accessed 15 Dec 2021
27. Nair V, Hinton G (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of ICML, vol 27. Haifa, Israel, pp 807–814

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Daofei Li received the B.S. degree in Vehicle Engineering from the Jilin University of Technology, Changchun, China, in 2003, and the Ph.D. degree in Vehicle Engineering from the Shanghai Jiao Tong University, Shanghai, China, in 2008.

From 2008 to 2010, he was a Post-Doc with the Institute for Power Machinery and Vehicular Engineering, Faculty of Engineering, Zhejiang University, Hangzhou, Zhejiang (PMVEI, ZJU). From 2010 to 2015, he was Assistant Professor with PMVEI, ZJU. In 2011, he was a Visiting Scholar with the University of Missouri-Columbia, and from 2014 to 2016, he was a Visiting Scholar with the University of Michigan, Ann Arbor, Michigan. Since 2016, he has been Associate Professor with PMVEI, ZJU. He currently directs the Research Group of Human-Mobility-Automation, and his research interests include vehicle dynamics and control, intelligent vehicle and complete human-vehicle system dynamics.

Ao Liu received the B.S. degree in Mechanical Design Manufacturing and Automation from Zhejiang University, Hangzhou, China, in 2017.

He is currently working toward the Doctor degree at the College of Energy Engineering, Zhejiang University, Hangzhou, Zhejiang. His research topics are personalized driving algorithms, intelligent vehicle and machine learning.