Engineering 33 (2024) 108–120

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Safety for Intelligent and Connected Vehicles—Article

Evolutionary Decision-Making and Planning for Autonomous Driving Based on Safe and Rational Exploration and Exploitation

Kang Yuan [a,b], Yanjun Huang [c,*], Shuo Yang [c], Zewei Zhou [c], Yulei Wang [a], Dongpu Cao [d], Hong Chen [a,*]

[a] College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China
[b] Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 201210, China
[c] School of Automotive Studies, Tongji University, Shanghai 201804, China
[d] School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 5 August 2022
Revised 1 February 2023
Accepted 26 March 2023
Available online 22 June 2023

Keywords:
Autonomous driving
Decision-making
Motion planning
Deep reinforcement learning
Model predictive control

ABSTRACT

Decision-making and motion planning are extremely important in autonomous driving to ensure safe driving in a real-world environment. This study proposes an online evolutionary decision-making and motion planning framework for autonomous driving based on a hybrid data- and model-driven method. First, a data-driven decision-making module based on deep reinforcement learning (DRL) is developed to pursue a rational driving performance as much as possible. Then, model predictive control (MPC) is employed to execute both longitudinal and lateral motion planning tasks. Multiple constraints are defined according to the vehicle's physical limit to meet the driving task requirements. Finally, two principles of safety and rationality for the self-evolution of autonomous driving are proposed. A motion envelope is established and embedded into a rational exploration and exploitation scheme, which filters out unreasonable experiences by masking unsafe actions so as to collect high-quality training data for the DRL agent. Experiments with a high-fidelity vehicle model and MATLAB/Simulink co-simulation environment are conducted, and the results show that the proposed online-evolution framework is able to generate safer, more rational, and more efficient driving action in a real-world environment.

1. Introduction

Autonomous vehicles are a product of the deep integration of the automotive industry with new-generation information technologies such as artificial intelligence (AI) and high-performance computing, and they represent one of the most important directions of global automotive development. Decision-making and motion planning are the core of autonomous driving, as they directly determine how an autonomous vehicle moves and reacts to its dynamic environment. The decision-making module receives environment and vehicle information, and outputs the desired driving behavior to the motion planning module. The latter further outputs the desired trajectory to the trajectory tracker or directly outputs the desired commands to the vehicle actuators. Thus, these two modules form the "brain" of an autonomous vehicle, and their performance directly affects the ability of the vehicle to deal with a dynamic and open traffic environment.

The decision-making methods used in autonomous driving include rule-, optimization-, utility-function-, and AI-based methods. Rule-based methods are simple, but their applicable scenarios are limited. Nilsson et al. [1] investigate rules to determine the appropriate lane-changing time by choosing a safe trajectory through longitudinal planning. Another typical method based on a hierarchical state machine is also widely used [2]. Noh [3] proposes a robust method using risk metrics and Bayesian networks, with a distributed reasoning structure to ensure safety. Optimization-based methods can achieve optimality, but it is difficult to use them to handle model-free problems. Nilsson and Sjöberg [4] employ a hybrid logic system to develop an integrated decision-making method based on model predictive control (MPC), and predicting the movement of surrounding vehicles is further considered in Refs. [5,6]. Karlsson et al. [7] first use MPC to generate candidate trajectories, and then determines the optimal decision through optimization. Nilsson et al. [8] consider the average travel time, remaining time, and traffic rules to form a utility

* Corresponding authors.
E-mail addresses: yanjun_huang@tongji.edu.cn (Y. Huang), chenhong2019@tongji.edu.cn (H. Chen).

https://doi.org/10.1016/j.eng.2023.03.018

function to generate the target lane. Cui et al. [9] consider the gap and speed satisfaction from the preceding vehicle to calculate a utility value. Comfort, efficiency, safety, and human-like lane selection probability are also considered to design a utility function in Ref. [10]. Utility-function methods have simple structures, but the selection of evaluation indicators is complicated.

For motion planning, two main frameworks are popularly used in the literature. In the first framework, the trajectory is planned first, and then trajectory tracking is executed. Common trajectory planning methods include the polynomial method [11], the spline method [12], and the clothoid method [13]. Widely used trajectory tracking methods include proportional–integral–derivative (PID) control [14,15], sliding mode control (SMC) [16,17], and MPC [18,19]. Most of these methods ignore the coupling between trajectory planning and tracking control, which can easily cause conflicts. For example, in a fast-changing environment, the planned trajectory may not be trackable. In the other framework, the trajectory planning stage is skipped by means of optimization methods, and steering and longitudinal-control commands are directly outputted to vehicle actuators. This framework can obtain optimality under certain conditions and considers the coupling between trajectory planning and tracking control. As a result, it has been widely studied, especially in regard to MPC-related methods [20–24]. MPC can be used to deal with the optimal control problem with multiple constraints and can naturally imitate a driver's predictive behavior.

In addition, with the recent development of AI methods, learning-based autonomous driving techniques have been widely studied [25] and are usually based on imitation learning (IL) [26] and reinforcement learning (RL) [27] methods. Liu et al. [28] use a Gaussian kernel support vector machine (SVM) to make lane-changing decisions, while Wang et al. [29] utilize long short-term memory (LSTM) to make human-like decisions. End-to-end learning is another popular technique that maps sensing information to the vehicle control commands. Xiao et al. [30] present a multi-modal conditional IL (CIL) method for end-to-end autonomous driving. Menner et al. [31] propose that the parameterized motion planning objective be learned via inverse learning. Regarding RL methods, Peng et al. [32] employ a dueling double deep Q-network (DQN) approach to design the steering controller. Lin et al. [33] further utilize a deep deterministic policy gradient (DDPG) algorithm for continuous adaptive cruise control. In addition, He et al. [34] present a constrained robust actor-critic (AC) method for lane changing under traffic uncertainties. Although AI-based methods are good at learning, they are highly dependent on data, and cannot easily ensure safety for safety-critical systems. Similar studies can also be found in Refs. [35–39] and others.

Autonomous vehicles are a typical safety-critical system, so the aforementioned methods still present challenges, especially in an open driving environment. First, from the perspective of system development, rule- and model-based methods are mostly used to develop decision-making and planning algorithms, or AI-based methods are employed to train feasible policies offline in the developing stage. These algorithms or policies are then deployed in autonomous vehicles, making it difficult to endow vehicles with online learning and continuous evolution in the operating stage with the driver and passengers in the loop. However, such a capability is extremely necessary in order for autonomous vehicles to be able to deal with an unknown, dynamic and open traffic environment in a continuable, growable, and reliable manner.

Second, the "black-box" nature of deep learning and the random trial-and-error mechanism of deep RL (DRL) seriously affect safety and trustworthiness when exploring and exploiting policy online in the operating stage. Therefore, it is another important challenge to realize the safe and rational evolution of autonomous driving. Finally, most existing studies ignore the mutual coupling between decision-making and motion planning, and design the two layers separately to achieve individual objectives in a sequential manner. These studies usually develop the decision-making layer without considering the planning capability boundary constrained by vehicle kinematics and dynamics. This will result in decisions that are too aggressive to be well executed by the planning layer or too conservative to waste the planning layer's capability; thus, such methods cannot achieve optimal performance of the whole decision-making and planning system. The present study addresses these problems, and its main contributions are as follows:

(1) A novel online-evolution framework of decision-making and planning for autonomous driving in the operating stage is proposed by developing a hybrid data- and model-driven method based on DRL and MPC. This framework takes advantage of the high self-adaptation and self-learning capabilities of data-driven methods, as well as the interpretability and ability to handle hard constraints of model-driven methods.

(2) Two principles for safety and rationality in the online evolution of autonomous driving are proposed. Based on the above framework, a safe-driving envelope is established, and a rational exploration and exploitation scheme is designed that filters out random and unsafe experiences by masking unsafe actions in order to obtain high-quality training data and realize the safe and rational self-evolution of autonomous driving.

(3) Mutual coupling between the decision-making layer and the planning layer is considered in order to pursue the optimal performance of the whole system. Based on a safe online-learning mechanism, the continuous evolution of the system within the capability boundary of the planning layer is realized, along with the maximum utilization of the capabilities of the planning layer.

The remainder of this paper is organized as follows: Section 2 introduces the whole proposed framework. The data-driven evolutionary decision-making module is then presented in Section 3 with the DQN problem formulation and parameter design. A model-driven motion planning method using MPC is elaborated in Section 4. Section 5 develops the safe and rational exploration and exploitation mechanism based on a predictive safe driving envelope model and a rational exploration and exploitation scheme. Finally, Section 6 demonstrates a case study, and Section 7 presents conclusions and identifies future work.

2. Proposed framework

Autonomous vehicles require not only high adaptability and learning ability in an open traffic environment but also strict safety and strong rationality. Data-driven methods are difficult to interpret and cannot ensure strict safety, although they are good at learning. In comparison, model-driven methods lack self-adaptation and self-learning, but they are interpretable and can handle various constraints. Therefore, this study proposes a framework by designing a hybrid data- and model-driven method to deal with decision-making and motion planning as a whole. Decision-making is more relevant to vehicle adaptation and learning capabilities, while motion planning is directly related to vehicle safety. In addition, to realize safe and rational self-evolution in a real-world driving environment, the framework introduces a safe-driving envelope to address difficult safety constraints, as well as a rational exploration and exploitation scheme. The proposed framework is capable of considering the coupling between decision-making and motion planning—that is, the evolution of decision-making is based on the capability boundary of motion planning.

RL has the advantages of being model-free, unsupervised, and a form of autonomous learning, and it is very suitable for learning to

make decisions that are difficult to model in complex uncertain scenarios. MPC has an inherent advantage in dealing with predictive optimization problems with hard constraints and well reflects the predictive driving behavior of human drivers. Therefore, in this study, we chose the DQN approach in DRL for learning discrete decision policies, while using MPC for safe motion planning. It should be noted that the planning layer in this study receives the decision commands and directly outputs the desired steering and acceleration commands to the vehicle actuators. The overall structure proposed in this study is shown in Fig. 1.

This framework consists of an environment module, a data-driven evolutionary decision module, a model-driven motion planning module, and a safe and rational policy exploration and exploitation module. First, the environment module outputs the motion states of the ego vehicle and the surrounding vehicles to the three modules. Then, in the decision module, the DQN agent learns iteratively by continuously interacting with the environment through trial-and-error. At each time step, the DQN agent outputs the decision command to the motion planning module and receives a filtered signal on the current bad decision from

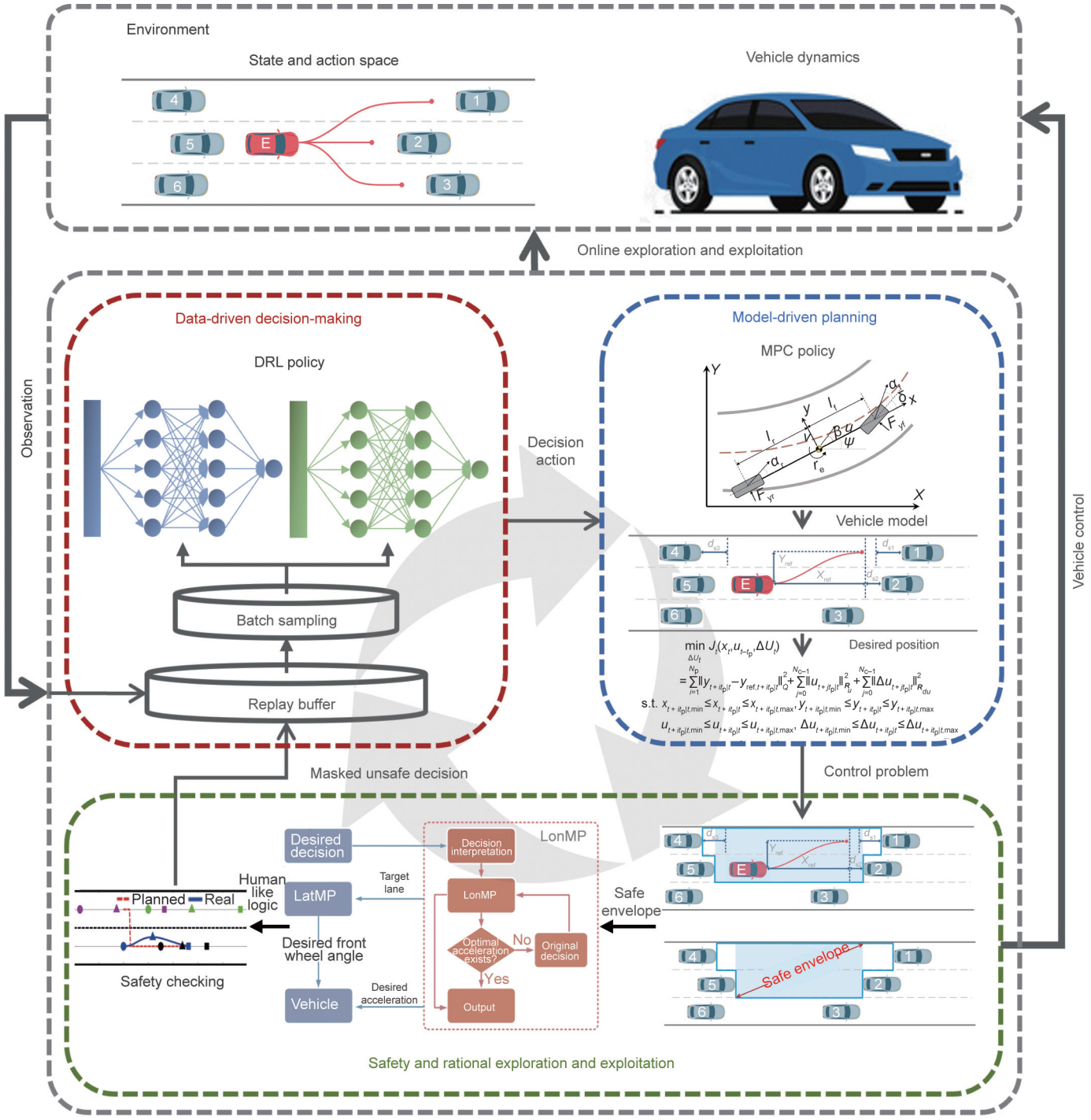

Fig. 1. The overall framework of online evolutionary decision-making and motion planning for autonomous driving in the operating stage. The formulas and parameters in the figure are defined in Section 4. 1–6: the six traffic vehicles; E: the ego vehicle; LatMP: the lateral motion planning module; LonMP: the longitudinal motion planning module.

the safe and rational policy exploration and exploitation module. Next, the MPC-based motion planning module decouples the desired driving behavior into longitudinal and lateral motions after receiving the decision commands. Accordingly, longitudinal and lateral planning are respectively performed. Finally, in the safe and rational policy exploration and exploitation module, a safe driving envelope is constructed to constrain the MPC planning problem. The rational exploration and exploitation scheme based on trial-and-error is further designed to perform safety checking and rational motion correction control of the desired decision commands in real time. In this way, the current unreasonable desired decisions are masked and fed back to the DQN decision agent. Meanwhile, the corresponding obtained acceleration and steering commands are outputted to the environment to control the ego vehicle.

3. Data-driven evolutionary decision-making

This section introduces the DQN-based evolutionary decision module, including decision problem formulation and parameter design.

3.1. DQN-based decision-making problem formulation

The decision-making of autonomous driving can be considered to be a sequential optimal decision-making process, which can be described by a Markov decision process (MDP). Based on the MDP, the RL agent can guide the autonomous vehicle to interact and learn with the environment by defining the reward function, constructing the optimization objective, and using the trial-and-error learning mechanism, finally obtaining the optimal decision-making policy. An MDP is defined as a tuple $M = \langle S, A, T, R, \gamma \rangle$, where S is the state space; A is the action space; $T = \left\{ p^{a_t}_{s_t s_{t+1}} : s_t, s_{t+1} \in S, a_t \in A \right\}$ is the state transition probability from state s_t to s_{t+1} with action a_t (where $p^{a_t}_{s_t s_{t+1}}$ is the state transition probability and a_t is the chosen action); $R = r^{a_t}_{s_t s_{t+1}}$ is the instant reward of the above state transition and r is the reward value; and γ is the discount factor. The corresponding decision-making policy is defined as $\pi(a_t|s_t) = p^{a_t}_{s_t}$, denoting the probability of choosing a_t at s_t. The state value function $V_\pi(s_t)$ is the expected accumulated reward obtained by executing π starting from s_t, which is defined as follows:

$$V_\pi(s_t) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r^{a_{t+k}}_{s_{t+k} s_{t+k+1}} | s_t \right] \tag{1}$$

where E_π is the calculation of expectation, k is the state transition index, and t is the current time step.

The state-action value function $Q_\pi(s_t, a_t)$ is defined as follows:

$$Q_\pi(s_t, a_t) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r^{a_{t+k}}_{s_{t+k} s_{t+k+1}} | s_t, a_t \right] \tag{2}$$

The optimal policy of the agent is a policy that enables every state with its maximum state value, which is defined as:

$$\pi^* = \arg\max_{\pi} V_\pi(s_t), \forall s_t \in S \tag{3}$$

where π^* is the optimal policy.

The optimal policy ensures the unique optimal state value $V^*(s_t)$ and state-action value $Q^*(s_t, a_t)$ of every state and state-action pair, respectively, which can be calculated by solving the Bellman optimality equation (BOE):

$$\begin{cases} V^*(s_t) = \max_{a_t} \sum_{s_{t+1}} p^{a_t}_{s_t s_{t+1}} [r^{a_t}_{s_t s_{t+1}} + \gamma V^*(s_{t+1})] \\ Q^*(s_t, a_t) = \sum_{s_{t+1}} p^{a_t}_{s_t s_{t+1}} [r^{a_t}_{s_t s_{t+1}} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})] \end{cases} \tag{4}$$

The optimal action $\pi^*(a_t|s_t)$ can be calculated as follows:

$$\pi^*(a_t|s_t) = \operatorname*{argmax}_{a_t \in A} Q^*(s_t, a_t), \forall s_t \in S \tag{5}$$

However, it is difficult to solve the BOE directly when the dimension of S or A is too large. Q-learning is a classical off-policy RL algorithm based on temporal difference (TD), which solves the BOE by the approximate state-action value iteration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r^{a_t}_{s_t s_{t+1}} + \gamma \max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{6}$$

where α is the RL learning rate. Based on Q-learning, DQN utilizes a neural network to approximate the state-action value function so as to train the RL agent with a continuous state space such as the decision-making agent in autonomous driving. Similarly, the parameter of the Q-network θ_t can be updated as follows:

$$\begin{aligned} \theta_{t+1} = \theta_t + \alpha [r^{a_t}_{s_t s_{t+1}} + \gamma \max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1}; \theta_{\text{target}}) \\ - Q(s_t, a_t; \theta_t)] \times \nabla Q(s_t, a_t; \theta_t) \end{aligned} \tag{7}$$

where θ_{target} is the parameter of the target Q-network. Furthermore, double-DQN is proposed to improve the overfitting of traditional DQN [36], which updates θ_t as follows:

$$\begin{aligned} \theta_{t+1} = \theta_t + \alpha [r^{a_t}_{s_t s_{t+1}} + \gamma Q(s_{t+1}, \operatorname*{argmax}_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1}; \theta_t); \theta_{\text{target}}) \\ - Q(s_t, a_t; \theta_t)] \times \nabla Q(s_t, a_t; \theta_t) \end{aligned} \tag{8}$$

3.2. DQN parameter design

3.2.1. State and action space configuration

The action space in autonomous driving must accurately and completely describe the driving behavior during a specific driving task. On the one hand, the selection of the state space must consider the most important environment elements that induce driving behavior; on the other hand, it is necessary to ignore less important environment elements as much as possible to reduce their interference on driving behavior, which can simultaneously reduce the dimension of the state space to save computation resources. In this study, we design an algorithm for the driving task of an ego vehicle traveling in three-lane traffic flow. The driving behaviors in this task include lane keeping, left lane changing, and right lane changing, which can be regarded as behaviors that pursue different target lanes. Therefore, the identifiers (IDs) of all the possible target lanes are directly selected to construct the action space. The perception attention of a human driver directly affects the generation of that person's driving behavior and is the most important factor inducing different driving behaviors. In general, the most basic perception attention range of a human driver can be described by the position and motion information of the ego vehicle and its nearest surrounding vehicles; thus, this information is selected to construct the state space. It should be noted that this state space construction method can be extended to other driving tasks, as well as to a wider range of human-like perception. The state space S and the action space A are depicted in Fig. 2, and are defined as follows:

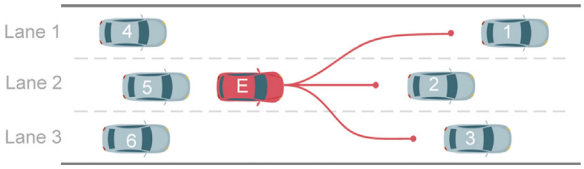

Fig. 2. The state and action space configuration.

$$\begin{cases} S = \left[v_x, v_y, Y, \Delta d_1, \Delta v_{x1}, \Delta d_2, \Delta v_{x2}, \Delta d_3, \Delta v_{x3}, \Delta d_4, \Delta v_{x4}, \Delta d_5, \Delta v_{x5}, \Delta d_6, \Delta v_{x6}\right]^{\mathrm{T}} \\ A = \left[\mathrm{TL}_1, \mathrm{TL}_2, \mathrm{TL}_3\right]^{\mathrm{T}} \end{cases} \tag{9}$$

where S consists of the longitudinal and lateral velocities (v_x, v_y) of the ego vehicle in vehicle coordinates, the lateral position (Y) of the ego vehicle in global coordinates, and the relative distances ($\Delta d_1, \Delta d_2, \Delta d_3, \Delta d_4, \Delta d_5, \Delta d_6$) and velocities ($\Delta v_{x1}, \Delta v_{x2}, \Delta v_{x3}, \Delta v_{x4}, \Delta v_{x5}, \Delta v_{x6}$) between the ego vehicle and the six surrounding traffic vehicles; and the variables TL_1–TL_3 in A separately denote the three target lane IDs.

3.2.2. Reward function design

The reward function is the driving force that guides the learning direction of the RL agent. In autonomous driving, the primary criterion for the design of the reward function is to reflect the global goal of the driving task, which in our case is to make the ego vehicle pursue the highest possible traffic efficiency under the premise of the road speed limit. Therefore, this study considers the speed reward. It should be noted that this paper does not need to consider reward dimensions such as safety (i.e., collision avoidance) and comfort using traditional reward design methods; rather, it transfers these global goals to the model-driven planning layer and the safe and rational exploration and exploitation mechanism, to be introduced later. This mechanism is based on trial-and-error of the desired decision, where the success of trial-and-error means that the planning layer can immediately implement the safe planning to execute the desired decision command, and failed trials mean that the desired decision cannot be executed immediately in a safe way and will be masked and corrected back to safe controls. In this way, the mechanism can directly and stably ensure the most basic driving needs such as safety and comfort.

Therefore, the reward function of the autonomous driving RL agent is defined as follows:

$$r_{s_t s_{t+1}}^{a_t} = w_v \left(\frac{v_x - v_{\max}}{v_{\max}}\right)^2 \tag{10}$$

where $v_{\max}$ is the road speed limit and w_v is the weighting coefficient.

4. Model-driven motion planning

This section introduces the MPC-based motion planning module, including the MPC prediction model design and the motion planning problem formulation.

4.1. MPC prediction model design

4.1.1. Vehicle kinematics and dynamics modeling

In different driving tasks and even different driving behaviors, human drivers have different preferences for longitudinal and lateral motions. When solving the longitudinal and the lateral motion control problems of the ego vehicle through the same optimization problem with the same optimization objective, it is often difficult to reasonably allocate the priority of longitudinal and lateral optimal control, which results in unstable vehicle motion control. Therefore, this study decouples the longitudinal and lateral motion of the vehicle and uses MPC to control the vehicle.

The longitudinal differential kinematics model of the ego vehicle is defined as follows:

$$\begin{cases} \dot{X} = v_X \\ \dot{v}_X = a_X \end{cases} \tag{11}$$

where X, v_X, and a_X denote the longitudinal position, the longitudinal velocity, and the longitudinal acceleration of the ego vehicle in global coordinates, respectively.

To consider the lateral dynamics of the ego vehicle more accurately and to improve the lateral motion stability, a linear tire model is employed. This model assumes a linear relationship between the tire force and the slip angle over a certain range of tire slip angle and assumes a small front wheel steering angle. Based on this assumption, a linear MPC prediction model can be constructed, which can then be solved by means of standard quadratic programming, avoiding the high computation burden brought by nonlinear MPC calculations. This assumption can also be used for normal stable driving conditions. The lateral dynamics model of the ego vehicle is shown in Fig. 3.

$$\begin{cases} \dot{v} = \frac{(F_{yf}+F_{yr})}{m} - \frac{\dot{X}+v\sin\psi}{\cos\psi} r_e \\ \dot{\psi} = r_e \\ \dot{r}_e = \frac{(F_{yf}l_f - F_{yr}l_r)}{I_z} \\ \dot{Y} = \frac{\dot{X}+v\sin\psi}{\cos\psi}\sin\psi + v\cos\psi \end{cases} \tag{12}$$

where v is the lateral velocity in vehicle coordinates; r_e is the yaw rate; ψ represents the yaw angle in global coordinates; m and I_z denote the mass and the moment of inertia, respectively; l_f and l_r are the distances from the vehicle's center of gravity to the front and the rear axles, respectively; F_{yf} and F_{yr} are the lateral forces of the front and the rear tires, respectively; and $\dot{v}$, $\dot{\psi}$, $\dot{r_e}$, and $\dot{Y}$ denote the differential calculations. The linear tire forces can be calculated as follows:

$$\begin{cases} F_{yf} = C_{\alpha f}\left(\delta - \frac{(v+l_f r_e)\cos\psi}{\dot{X}+v\sin\psi}\right) \\ F_{yr} = C_{\alpha r}\left(-\frac{(v-l_r r_e)\cos\psi}{\dot{X}+v\sin\psi}\right) \end{cases} \tag{13}$$

where $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffnesses of the front and rear tires, respectively; and δ is the front wheel steering angle. It should be noted that the kinematics and dynamics models of the ego vehicle in this study can be directly transferred to the Frenet coordinate system through coordinate transformation.

4.1.2. Driving behavior interpretation

When using MPC to plan the motions of the desired driving behavior, another important issue is how to model the behavior as an optimization objective with constraints that MPC can

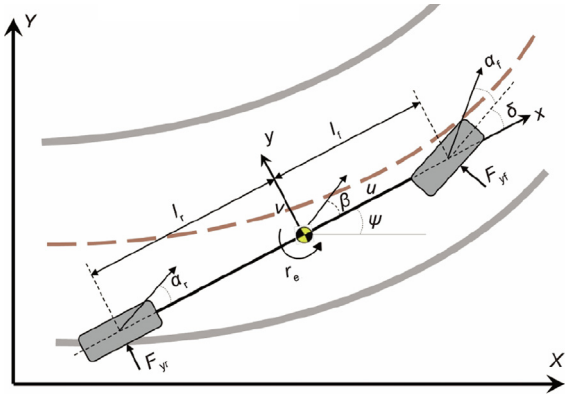

Fig. 3. The lateral dynamics model of the ego vehicle. F_{yf}, F_{yr}: the lateral forces of the front and the rear tires, respectively; α_f, α_r: the side slip angles of the front and the rear tires, respectively; l_f, l_r: the distances from the vehicle's center of gravity to the front and the rear axles, respectively; r_e: the yaw rate; u, v: the longitudinal and the lateral velocities in vehicle coordinates, respectively; ψ: the yaw angle in global coordinates; δ: the front wheel steering angle; β: the side slip angle of the vehicle's center of gravity; x and y separately denote the longitudinal and lateral axis in the vehicle coordinate system, respectively.

understand. Since the desired longitudinal and lateral positions of the ego vehicle reflect the most important characteristics of the lane-keeping and lane-changing behaviors, we select the longitudinal and lateral positions as state variables and carry out motion planning in the MPC prediction horizon to track the desired position signals that reflect the desired decision-making commands in real time.

In the prediction horizon, the state, output, and control vectors of the longitudinal and lateral motions are defined as follows:

$$\begin{cases} x_{\text{lon}} = [X, v_X]^{\text{T}}, y_{\text{lon}} = X, u_{\text{lon}} = a_X \\ x_{\text{lat}} = [\, v \quad \psi \quad r_e \quad Y\,]^{\text{T}}, y_{\text{lat}} = Y, u_{\text{lat}} = \delta \end{cases} \tag{14}$$

where x_{lon}, y_{lon}, and u_{lon} are the longitudinal model variables; and x_{lat}, y_{lat}, and u_{lat} belong to the lateral model. The time-discrete prediction model with a longitudinal time step $t_{\text{p,lon}}$ and a lateral step $t_{\text{p,lat}}$ is defined as follows:

$$\begin{cases} x_{\text{lon},k_{\text{lon}}+1} = \boldsymbol{A}_{\text{lon}} x_{\text{lon},k_{\text{lon}}} + \boldsymbol{B}_{\text{lon}} u_{\text{lon},k_{\text{lon}}} + \boldsymbol{C}_{\text{lon}} \\ y_{\text{lon},k_{\text{lon}}+1} = \boldsymbol{C}_{c,\text{lon}} x_{\text{lon},k_{\text{lon}}+1}, k_{\text{lon}} = 0, 1, ..., N_{\text{p,lon}} - 1 \\ x_{\text{lat},k_{\text{lat}}+1} = \boldsymbol{A}_{\text{lat}} x_{\text{lat},k_{\text{lat}}} + \boldsymbol{B}_{\text{lat}} u_{\text{lat},k_{\text{lat}}} + \boldsymbol{C}_{\text{lat}} \\ y_{\text{lat},k_{\text{lat}}+1} = \boldsymbol{C}_{c,\text{lat}} x_{\text{lat},k_{\text{lat}}+1}, k_{\text{lat}} = 0, 1, ..., N_{\text{p,lat}} - 1 \end{cases} \tag{15}$$

where k_{lon} and k_{lat} are time indexes in longitudinal and lateral predictions, respectively; $\boldsymbol{A}_{\text{lon}}$, $\boldsymbol{B}_{\text{lon}}$, $\boldsymbol{C}_{\text{lon}}$, $\boldsymbol{C}_{c,\text{lon}}$, $\boldsymbol{A}_{\text{lat}}$, $\boldsymbol{B}_{\text{lat}}$, $\boldsymbol{C}_{\text{lat}}$, and $\boldsymbol{C}_{c,\text{lat}}$ are system matrixes; $N_{\text{p,lon}}$ and $N_{\text{p,lat}}$ are the longitudinal and lateral prediction horizons, and

$$\boldsymbol{A}_{\text{lon}} = \begin{bmatrix} 1 & t_{\text{p,lon}} \\ 0 & 1 \end{bmatrix}, \boldsymbol{B}_{\text{lon}} = \begin{bmatrix} 0 \\ t_{\text{p,lon}} \end{bmatrix}, \boldsymbol{C}_{\text{lon}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{C}_{c,\text{lon}} = [1, 0] \tag{16}$$

The matrixes $\boldsymbol{A}_{\text{lat}}$, $\boldsymbol{B}_{\text{lat}}$, $\boldsymbol{C}_{\text{lat}}$, and $\boldsymbol{C}_{c,\text{lat}}$ can be calculated by linearizing the lateral nonlinear dynamics model of the ego vehicle [22].

The desired longitudinal and lateral positions of different driving behaviors are shown in Fig. 4. The lane width W_{lane} is 4 m in this study. The desired longitudinal position is determined based on a safe distance from all the front vehicles. The desired lateral position comes from the decision-making command, which is also the lateral position of the centerline of the target lane. In the prediction horizon, the desired longitudinal and lateral positions can be defined as follows:

$$\begin{cases} X_{\text{ref},k_{\text{lon}}+1} = X_{\text{ref},t} \\ Y_{\text{ref},k_{\text{lat}}+1} = Y_{\text{ref},t} \end{cases} \tag{17}$$

where $X_{\text{ref},t}$ and $X_{\text{ref},t}$ are the desired positions at t.

4.2. MPC-based motion planning problem formulation

4.2.1. Optimization problem statement

The objective function of MPC-based motion planning in this study is defined as follows:

$$\begin{aligned} J_t(x_t, u_{t-t_{\text{p}}}) = & \sum_{i=1}^{N_{\text{p}}} \left\| y_{t+it_{\text{p}}|t} - y_{\text{ref},t+it_{\text{p}}|t} \right\|_{\boldsymbol{Q}}^2 + \sum_{j=0}^{N_{\text{c}}-1} \left\| u_{t+jt_{\text{p}}|t} \right\|_{\boldsymbol{R}_u}^2 \\ & + \sum_{j=0}^{N_{\text{c}}-1} \left\| \Delta u_{t+jt_{\text{p}}|t} \right\|_{\boldsymbol{R}_{du}}^2 \end{aligned} \tag{18}$$

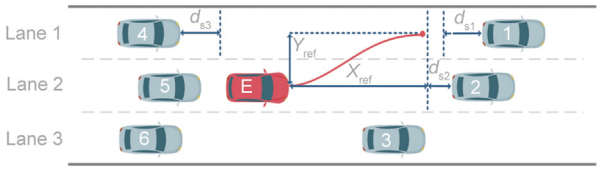

Fig. 4. Interpretation of driving behavior. d_{s1}, d_{s2}, and d_{s3}: the safe distances; X_{ref}, Y_{ref}: the desired positions.

where t_{p} is the prediction time step; N_{p} is the prediction horizon; N_{c} is the control horizon; $\boldsymbol{Q}$ is the weighting matrix to pursue the desired driving decision command; and $\boldsymbol{R}_u$ and $\boldsymbol{R}_{du}$ are the weighting matrixes to minimize the control and its jerk, respectively, which are related to driving comfort. On this basis, the constrained MPC optimization problem is formulated as follows:

$$\begin{aligned} & \min_{\Delta \boldsymbol{U}_t} J_t(x_t, u_{t-t_{\text{p}}}, \Delta \boldsymbol{U}_t) \\ & \text{s.t. } x_{t+it_{\text{p}}|t,\min} \le x_{t+it_{\text{p}}|t} \le x_{t+it_{\text{p}}|t,\max} \\ & y_{t+it_{\text{p}}|t,\min} \le y_{t+it_{\text{p}}|t} \le y_{t+it_{\text{p}}|t,\max} \\ & u_{t+it_{\text{p}}|t,\min} \le u_{t+it_{\text{p}}|t} \le u_{t+it_{\text{p}}|t,\max} \\ & \Delta u_{t+it_{\text{p}}|t,\min} \le \Delta u_{t+it_{\text{p}}|t} \le \Delta u_{t+it_{\text{p}}|t,\max} \end{aligned} \tag{19}$$

where $\Delta \boldsymbol{U}_t$ is the control vector and

$$\begin{cases} \Delta \boldsymbol{U}_t = \left[\Delta u_{t|t}, \quad \Delta u_{t+t_{\text{p}}|t}, \ldots, \quad \Delta u_{t+\left(N_{\text{p}}-1\right)t_{\text{p}}|t} \right]^{\text{T}} \\ \Delta u_{t+jt_{\text{p}}|t} = u_{t+jt_{\text{p}}|t} - u_{t+(j-1)t_{\text{p}}|t} \end{cases} \tag{20}$$

The above problem is then applied to the longitudinal and the lateral motion planning, respectively, of the ego vehicle.

4.2.2. Constraints setting

Autonomous driving must satisfy physical constraints related to the physical performance of the vehicle and task constraints related to driving tasks, so as to achieve safe, comfortable, and stable driving. The physical constraints include the acceleration-related constraints, constraints related to the front wheel steering angle, and the sideslip angle constraints of the tire model. Substituting a_X, δ, α_{f}, and α_{r} into Eqs. (19) and (20), the upper and lower bounds are defined as follows:

$$\begin{cases} a_{X,t+it_{\text{p,lon}}|t,\min} \le a_{X,t+it_{\text{p,lon}}|t} \le a_{X,t+it_{\text{p,lon}}|t,\max} \\ \Delta a_{X,t+it_{\text{p,lon}}|t,\min} \le \Delta a_{X,t+it_{\text{p,lon}}|t} \le \Delta a_{X,t+it_{\text{p,lon}}|t,\max} \\ \delta_{\min} \le \delta_{t+jt_{\text{p,lat}}|t} \le \delta_{\max} \\ \Delta \delta_{\min} \le \Delta \delta_{t+jt_{\text{p,lat}}|t} \le \Delta \delta_{\max} \\ \alpha_{\text{f},\min} \le \alpha_{f,t+it_{\text{p,lat}}|t} \le \alpha_{\text{f},\max} \\ \alpha_{\text{r},\min} \le \alpha_{r,t+it_{\text{p,lat}}|t} \le \alpha_{\text{r},\max} \end{cases} \tag{21}$$

where $t_{\text{p,lon}}$ and $t_{\text{p,lat}}$ are longitudinal and lateral prediction time steps, respectively.

It is notable that the above jerk constraints on the acceleration and the front wheel angle can also be regarded as comfort and stability constraints, from the perspective of driving tasks. The other comfort constraint is the lateral acceleration constraint, which is defined as follows:

$$ay, t + it_{\text{p,lat}}|t_{y,\max y,\min} \tag{22}$$

In addition to the above constraints, traffic rules are considered. Thus, the road speed limit constraint is defined as follows:

$$vX, t + it_{\text{p,lon}}|t_{X,t+it_{\text{p,lon}}|t,\max X,t+it_{\text{p,lon}}|t,\min} \tag{23}$$

The aim is to imitate the comfortable acceleration and deceleration behavior of human drivers. Thus, in the process of acceleration, the changing rate of the acceleration gradually decreases with the increase in acceleration, while in the process of deceleration, the changing rate of the acceleration gradually decreases with the decrease in acceleration. This avoids the discomfort caused by continuously stepping on the accelerating or braking pedal. Therefore, the acceleration jerk constraint is defined as follows:

$$\begin{cases} \Delta a_{X,t+it_{\text{p,lon}}|t,\min} = w_1 \left(a_{X,t+it_{\text{p,lon}}|t,\min} - a_X \right) \\ \Delta a_{X,t+it_{\text{p,lon}}|t,\max} = w_2 \left(a_{X,t+it_{\text{p,lon}}|t,\max} - a_X \right) \end{cases} \tag{24}$$

where w_1 and w_2 are weighting coefficients.

In addition, the longitudinal and lateral safety constraints of driving behavior determine the safety of the entire decision-making and planning system; they are also the safety basis for online trial-and-error by the RL agent. This part will be introduced in the next section.

5. Safe and rational exploration and exploitation

To realize the online evolution of autonomous driving in the operating stage, the safety and the rationality of driving exploration and exploitation are two major principles that must be obeyed. These principles are the key factors that affect the safety, comfort, and trust of the driver and passengers in online autonomous driving. This section introduces the corresponding modeling methods for these two principles, including predictive safe-driving envelope modeling and a rational exploration and exploitation scheme.

5.1. A predictive safe-driving envelope

Safety is the primary principle. To ensure the strict safety of the autonomous vehicle during its evolution, this study proposes the introduction of longitudinal and lateral safety requirements of driving behavior into the MPC problem in the form of hard constraints. These safety constraints are modeled as a predictive safe-driving envelope, as shown in Fig. 5, where the blue areas represent the safe spaces for the ego vehicle in different lanes, and the longitudinal envelope boundary is determined according to the shorter one of the safe spaces in different lanes.

Therefore, the longitudinal and lateral position constraints in the MPC prediction horizon are defined as follows:

$$\begin{cases} X_{t+it_{\mathrm{p,lon}}|t,\min} \le X_{t+it_{\mathrm{p,lon}}|t} \le X_{t+it_{\mathrm{p,lon}}|t,\max} \\ Y_{\min} \le Y_{t+it_{\mathrm{p,lat}}|t} \le Y_{\max} \end{cases} \tag{25}$$

5.2. A rational exploration and exploitation scheme

Based on the safe envelope, this study further proposes a rational exploration and exploitation mechanism based on trial-and-error of the desired decision. In this mechanism, the success of the trial-and-error means that the planning layer can immediately implement safe planning to execute the desired decision; otherwise, failed trials mean that the desired decision cannot be executed immediately in a safe way. The working principle is shown in Fig. 6.

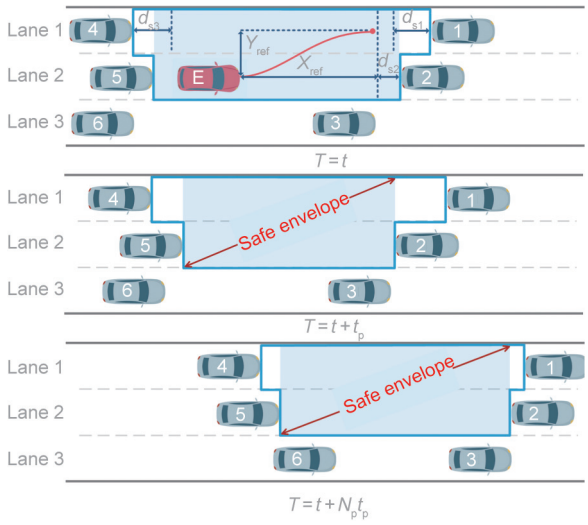

Fig. 5. The predictive safe-driving envelope.

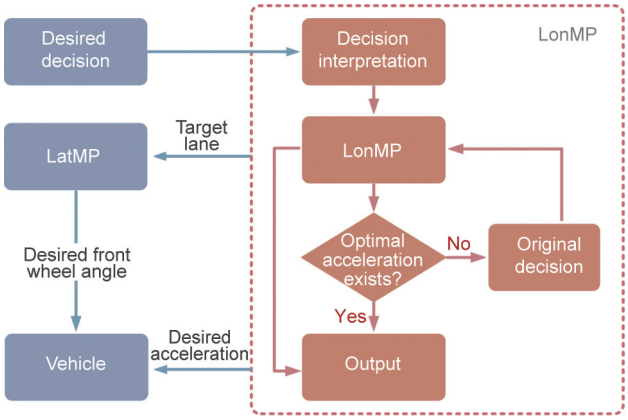

Fig. 6. The rational exploration and exploitation scheme.

It is only when the longitudinal motion planning of the desired decision has a feasible solution that the optimized longitudinal acceleration is executable; then, the corresponding desired lane centerline can be pursued through lateral motion planning. Otherwise, the longitudinal motion planning of the original decision (e.g., lane keeping) will be conducted and executed, and the lateral motion planning will lead the ego vehicle to track the original lane centerline. More specifically, the rational exploration and exploitation module is designed based on the criterion of longitudinal driving planning priority. This is intended to mimic the driving habits of humans. Usually, human drivers will prioritize the estimation of the longitudinal safety of the vehicle motion. If the decision made does not affect the longitudinal safety, then the decision can be used as one of the candidate decisions; if the decision affects the longitudinal safety, then the decision will not be executed. For example, when the vehicle is in the lane-keeping state, if it wants to change to the left lane, it can first use the MPC to carry out the longitudinal planning of the left lane changing behavior. If the longitudinal planning can obtain a reasonable optimal acceleration at the MPC algorithm level, this indicates that the optimal acceleration is safe and can be implemented; thus, the left lane change decision can be implemented. The optimal acceleration can be determined by whether the MPC has an optimal solution that satisfies the constraints. Furthermore, an acceleration limit interval related to comfort can be also introduced to evaluate qualities of MPC solutions. For example, if the deceleration obtained by the optimization is less than a certain threshold or the acceleration is greater than a certain threshold, then the ultimate optimal acceleration will not exist.

The proposed mechanism well describes a safe and rational online trial-and-error mechanism for the learning and evolution process of human drivers in the real world. For example, when driving, novice drivers usually constantly use trial-and-error driving behaviors and interact with surrounding vehicles so as to increase their driving experience and improve their driving ability. In this trial-and-error process, human drivers always attempt to drive their vehicle normally and stably to ensure safety, instead of driving by means of random unsafe exploration, as a traditional RL agent usually does. It should be noted that the "trial-and-error" in traditional RL refers to the agent utilizing randomization methods to enhance its exploration of unknown states or actions when making decisions, so as to increase the possibility of policy improvement. However, the goal of the trial-and-error mechanism of the MPC planning layer in this study is to imitate the process human drivers use to improve their driving skills.

6. Case study

In this section, the proposed method is verified by using a high-fidelity dynamics model, including a performance verification of the proposed framework and an analysis of the effects of key parameters.

6.1. Simulation setup

The Sim-to-Real problem is one of the challenges that restrict the extensive application of RL in real-world autonomous driving. This problem stems from the unreality of environment perception and vehicle dynamics in a simulation and training environment. Since the environment states are the relative positions and velocities of the surrounding vehicles rather than image information, this study uses a high-fidelity dynamics model and constructs a training environment using MATLAB/Simulink in order to truly reflect the dynamics of an autonomous vehicle in the real world as much as possible. This simulation scheme is capable of simulating the continuous learning and evolution process of autonomous driving in the real world.

In the RL algorithm, the replay buffer size is 100 000, the batch size is 32, the target network updating interval is 100, and the neural network is a fully connected network with a size of 16 × 50 × 50 × 1. Other simulation parameters are listed in Table 1.

6.2. Results and analysis

Here, we first verify the effectiveness of the algorithm in stable and unstable traffic flow in case 1, including the evolution performance and safety performance. Next, in cases 2 and 3, we discuss the effects of the driving style of the planning layer and the traffic flow density on the performance of the algorithm.

6.2.1. Case 1: Different average speeds of traffic flow

This case compares the effectiveness of the proposed framework for different average speeds of traffic flow. The results are provided in Fig. 7, where Figs. 7(a) and (b) show the results in stable traffic flow with an average speed of 20 (case 1-A) and 40 km·h^{-1} (case 1-B), respectively. The blue, orange, and yellow curves represent the average reward or speed within the past 1, 30, and 60 min, respectively. Fig. 7(c) further shows the results for the ego vehicle in an unstable traffic flow. The velocities of all the traffic vehicles are randomly assigned between 20 and 50 km·h^{-1}. The accelerations of the traffic vehicles are constrained to be within −4 to 2 m·s^{-2}, and are updated as follows:

$$a_{\mathrm{tv}} = \frac{(v_{\mathrm{tv,f}}{}^2 - v_{\mathrm{tv}}{}^2)}{2(X_{\mathrm{tv,f}} - X_{\mathrm{tv}} - d_{\mathrm{tv,s}})} \tag{26}$$

where X_{tv} and v_{tv} are the longitudinal position and the longitudinal velocity of the traffic vehicle, respectively; $X_{\mathrm{tv,f}}$ and $v_{\mathrm{tv,f}}$ denote the longitudinal position and velocity of the vehicle in front of the traffic vehicle; and $d_{\mathrm{tv,s}}$ refers to the safe distance. This setting is to simulate interactive driving behavior between vehicles in real traffic.

Because this paper focuses on the driving task in three-lane traffic flow, and its goal is to maximize the traffic efficiency as much as possible, the average speed is selected as the evaluation index. The impact of decision-making instructions on traffic efficiency is often in the long-term domain, reflecting the effect during a future period, and it is difficult to describe the quality of a certain decision using the instantaneous speed increase or decrease. Therefore, in this paper, we choose to evaluate and analyze the changes in the average speed within 1, 30, and 60 min, so as to reflect the degree of evolution of the decision-making layer. As can be seen from Figs. 7(a) and (b), with an increase in the training time, the average speed of the ego vehicle gradually increases and finally converges to a level about 10 km·h^{-1} above the designated traffic flow speed (20 and 40 km·h^{-1}). This means that, after online training, the ego vehicle has already learned how to obtain acceleration space through lane changing, so as to pursue a faster average speed. These results show the learning and evolution of the ego vehicle in its driving efficiency.

In this study, the acceleration capacity of the ego vehicle is determined by longitudinal motion planning based on MPC, and the maximum speed that can be achieved is determined by the average speed and density of the traffic flow. The higher the average speed of the traffic flow is and the lower the density is, the higher the equivalent longitudinal acceleration space of the ego vehicle will be, allowing the vehicle to achieve a higher average speed. Therefore, in this case, a stable traffic flow and fixed MPC parameters theoretically determine a maximum upper bound of the average speed in this environment. The goal of the DQN decision-making is to encourage the ego vehicle to continuously evolve to approach this maximum average speed, in theory. Therefore, in the simulation results, the MPC parameters, traffic flow characteristics, and DQN parameters jointly determine the level to which the ego vehicle can evolve, which in this case is to a level where the average speed exceeds the traffic flow speed by about 10 km·h^{-1}. As a further validation of the proposed framework in unstable traffic flow (case 1-C), Fig. 7(c) shows that, in the initial stage of training, the average reward curves of different time scales show a downward trend, because the agent prefers to explore to obtain more diverse experiences. With an increase in the training time, the average reward gradually increases and finally tends to stabilize. It is worth noting that the upper bound of the reward is determined by the average speed and density of the traffic flow, to a certain extent, and the trend of the reward curve is also greatly affected by the traffic flow.

For example, if a human driver encounters a traffic jam in a real traffic environment, regardless of how the driver drives, the speed will not be too high. During the training, the positions and velocities of traffic vehicles are random and time-varying, so the training environment is uncertain, resulting in oscillation of the reward curve. However, in general, with an increase in the training time, the agent shows an obvious upward trend in reward and achieves an obvious evolution. Consistent with the reward curve, the

Table 1
Simulation parameters.

Parameter	Value (units)	Parameter	Value (units)
γ	0.9	$R_{du,\mathrm{lon}}$	$0.1I$
α	0.001	$R_{du,\mathrm{lat}}$	0.1
w_v	−10 000	$V_{X,\min}$	0
$v_{\max}$	80 km·h^{-1}	$V_{X,\max}$	80 km·h^{-1}
m	1 270 kg	$a_{X,\min}$	−4 m·s^{-2}
l_{f}	1.015 m	$a_{X,\max}$	2 m·s^{-2}
l_{r}	1.895 m	w_1	5
I_z	1 536.7 kg·m^2	w_2	5
$C_{\alpha\mathrm{f}}$	45 860 N·rad^{-1}	$\delta_{\min}$	−15°
$C_{\alpha\mathrm{r}}$	25 796 N·rad^{-1}	$\delta_{\max}$	15°
$t_{\mathrm{p,lon}}$	0.05 s	$\Delta\delta_{\min}$	−0.75°
$t_{\mathrm{p,lat}}$	0.1 s	$\Delta\delta_{\max}$	0.75°
$N_{\mathrm{p,lon}}$	50	$\alpha_{\mathrm{f,min}}$	−5°
$N_{\mathrm{t,lat}}$	50	$\alpha_{\mathrm{f,max}}$	5°
$N_{\mathrm{c,lon}}$	2	$\alpha_{\mathrm{r,min}}$	−5°
$N_{\mathrm{c,lat}}$	2	$\alpha_{\mathrm{r,max}}$	5°
Q_{lon}	I	$Y_{\min}$	−6 m
Q_{lat}	I	$Y_{\max}$	6 m
$R_{u,\mathrm{lon}}$	$0.1I$	$a_{y,\min}$	$-0.4g$
$R_{u,\mathrm{lat}}$	$0.1I$	$a_{y,\max}$	$0.4g$

I: identity matrix; g: gravitational acceleration.

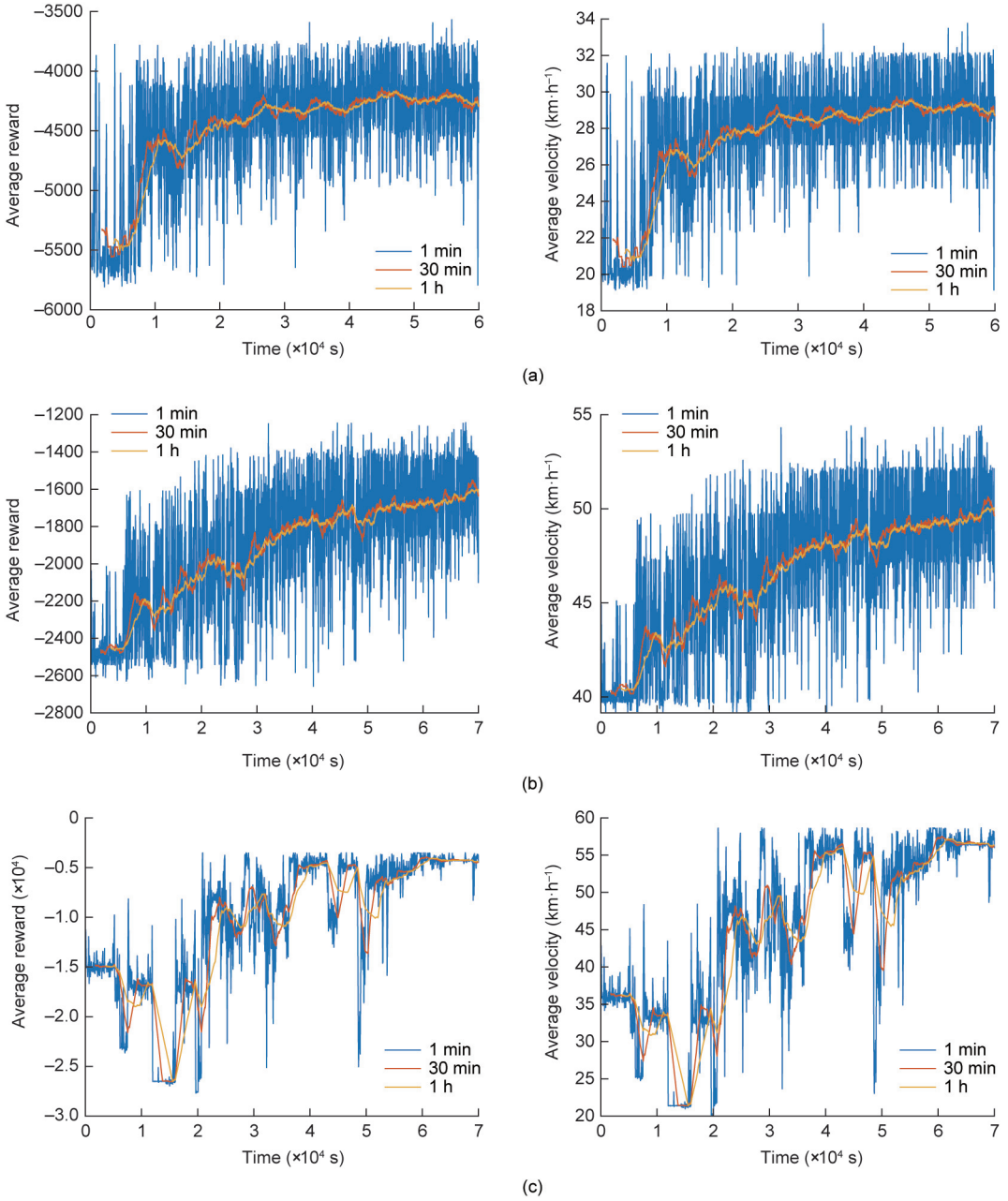

Fig. 7. The results of case 1: different average speeds of traffic flow. (a) Traffic speed: 20 km·h^{-1}; (b) traffic speed: 40 km·h^{-1}; (c) unstable traffic speed.

average longitudinal velocity of the ego vehicle decreases first and then increases. Since the velocities of all the traffic vehicles are less than 50 km·h^{-1}, it can be seen that the average velocity of the ego vehicle reaches about 57 km·h^{-1} in the later stage of training, which indicates that the ego vehicle has evolved to achieve a higher velocity through lane-changing behaviors in unstable traffic flow. This also reflects the online-evolution process of the ego vehicle.

To illustrate the safety performance, Fig. 8 presents the longitudinal distance from the ego vehicle to the front and rear traffic vehicles in the same lane, and the lateral position of the ego vehicle in the whole training process in case 1-A. Fig. 9 shows the longitudinal and lateral acceleration in the whole training process. It can be seen from Fig. 8 that, during the whole training process, the distance from the front traffic vehicle is always above 0 and the distance from the rear traffic vehicle is always below 0. This means that the ego vehicle does not collide with the traffic vehicles. Moreover, Fig. 8 shows that the ego vehicle never exceeds its lane boundaries (from −6 to 6 m). The distribution of vehicle's lateral positions changes with time, reflecting the trial-and-error behavior of multiple desired decisions. The above results all benefit from the safe driving envelope and the safe exploration and exploitation mechanism, which utilize MPC hard constraints to ensure safety. In addition, it can be seen from Fig. 9 that the longitudinal acceleration is constrained from −0.4*g* to 0.2*g* (where *g* is gravitational acceleration), and the lateral acceleration is always within 0.4*g* (mainly from −0.02*g* to 0.02*g*, which achieves reasonable comfort and stable longitudinal and lateral motion control.

As mentioned above, rational exploration and exploitation are embodied in two aspects in this study. First, traditional RL uses random exploration and repeated training. This is a mode of "knowing mistakes and making mistakes," which allows vehicle collisions. However, in autonomous driving, the online continuous training has zero tolerance for the collision safety problem, and the training cannot be reset back and forth. The proposed method,

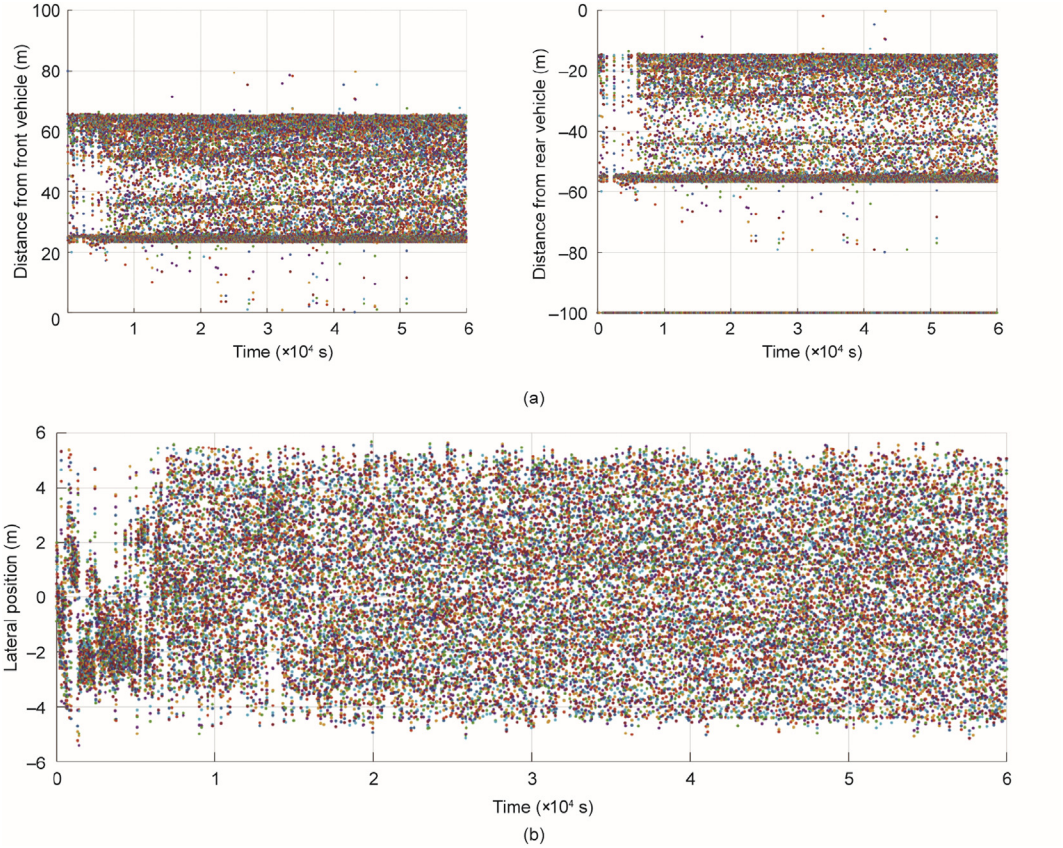

Fig. 8. Safety validation results. (a) Longitudinal distance from the ego vehicle to the front and rear vehicles in the same lane; (b) the lateral position in the whole training process.

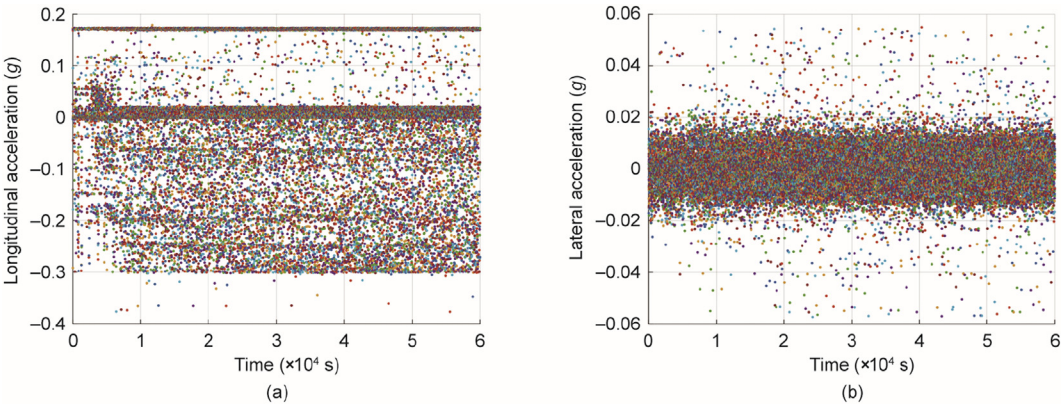

Fig. 9. (a) Longitudinal and (b) lateral acceleration in the whole training process.

which is an exploration mode of "knowing mistakes and correcting mistakes," is therefore needed. This mode simulates the learning process of human drivers. For example, when novice drivers learn lane-changing behavior, if they find that the lane-changing instructions they want to execute will lead to a collision, they will immediately give up on lane changing and return to lane keeping. Second, every driving experience collected in the training includes not only decision-making results but also planning results. The planning layer in this study uses MPC to naturally imitate the predicted driving behavior of human drivers. For example, in longitudinal motion planning, different following distances and their bounds reflect the following styles of conservative or aggressive drivers. This rational motion planning is the embodiment of the anthropomorphic nature of exploring and exploitation.

6.2.2. Case 2: Different planning layer styles

This case compares the effects of different MPC parameters of the planning layer on the proposed framework, which imitate conservative and aggressive human driving styles. The results are shown in Fig. 10. In case 2-A (shown in Fig. 10(a)), which imitates a conservative style, the longitudinal maximum position and the desired distance from the front vehicle in the MPC prediction horizon are 10 and 25 m, respectively. In case 2-B (shown in Fig. 10(b)), which represents an aggressive style, these parameters are 5 and 15 m, respectively. The average speed of the traffic flow is 40 km·h^{-1} in both cases 2-A and B.

Figs. 10(a) and (b) respectively correspond to the online training results of the ego vehicle based on a conservative and an aggressive MPC planning layer. In both cases, the ego vehicle can gradually

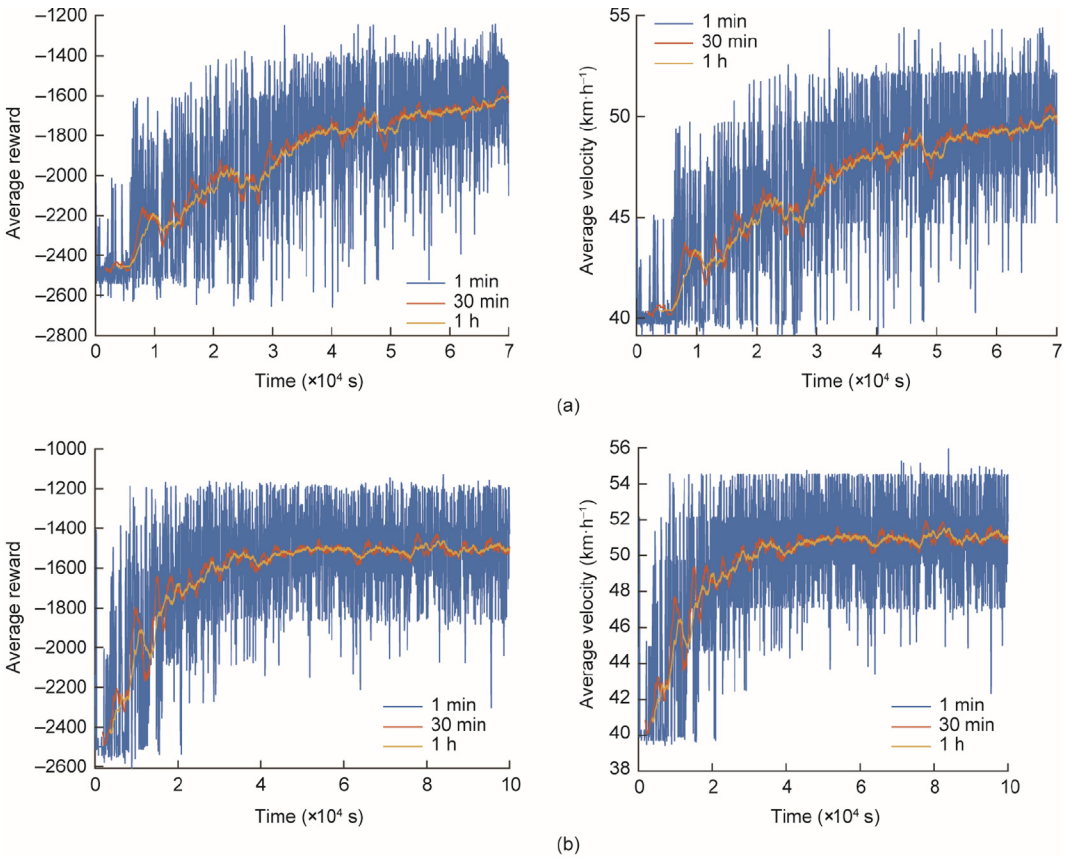

Fig. 10. Results of case 2: different planning layer styles. (a) Conservative planning layer based on MPC; (b) aggressive planning layer based on MPC.

evolve to exceed the average speed of the traffic flow by about 10 km·h^{-1}. However, the MPC planning layer in case 2-A has a larger longitudinal maximum position and a further desired position from the front vehicle than that in case 2-B, so its acceleration space is shorter. This causes a relatively conservative planning performance, causing the average speed of the ego vehicle with a more aggressive planning layer in case 2-B to converge faster, and the final speed is slightly higher than that in case 2-A. Therefore, a more aggressive planning level results in a higher maximum average speed being achieved by the whole decision-making and motion planning system. This is consistent with the driving performance of human drivers: The more aggressive drivers are, the higher their utilization of the traffic flow free space is, and the more they can pursue a higher average speed by changing lanes frequently.

6.2.3. Case 3: Different traffic flow densities

This case compares the effects of traffic flow density on the proposed framework. The results are shown in Fig. 11. The longitudinal distance between the traffic vehicles in the same lane are 80 and 120 m in cases 3-A and B (shown in Figs. 11(a) and (b)), respectively. The average speed of the traffic flow is 40 km·h^{-1} in both cases.

As shown in Fig. 11, the final average speed of the ego vehicle in case 3-A reaches 50 km·h^{-1}, while the final average speed in case 3-B is 55 km·h^{-1}. This finding shows that, at the same speed, sparse traffic flow allows the ego vehicle to evolve so that it can reach a higher average speed. This is because sparse traffic flow results in a greater distance between traffic vehicles, which provides a larger longitudinal acceleration space in which the ego vehicle can extend its acceleration time during lane keeping. It also increases the space for implementing lane-changing exploration. As a result, the probability of successful lane changing is increased, which permits the ego vehicle to extend the equivalent longitudinal space for acceleration through continuous lane-changing behavior. Taken together, these factors lead to an increase in the average speed of the ego vehicle.

7. Conclusions

This study deals with the online learning and evolution problem of decision-making and motion planning for autonomous driving in the operating stage by developing a hybrid data- and model-driven framework. This framework takes advantage of DRL's high self-learning capabilities and MPC's ability to deal with safety constraints and MPC's interpretability to develop a decision-making module and motion planner. The two principles of safety and rationality in the online evolution of autonomous driving in the operating stage are further proposed, and a corresponding safe and rational exploration and exploitation mechanism is designed. This mechanism is able to filter out random and unsafe experiences by masking unsafe actions so as to obtain high-quality training data with safe and human-like features. Moreover, based on the proposed framework, continuous evolution of the decision-making layer within the capability boundary of the planning layer is realized, along with the maximum utilization of the capabilities of the planning layer. Finally, safe and rational self-evolution of autonomous driving is realized. The results show that the proposed framework achieves the safe and rational online evolution of autonomous driving to pursue higher traffic efficiency. More specifically, it is found that ① the maximum speed that can be achieved is determined by the average speed and density of the

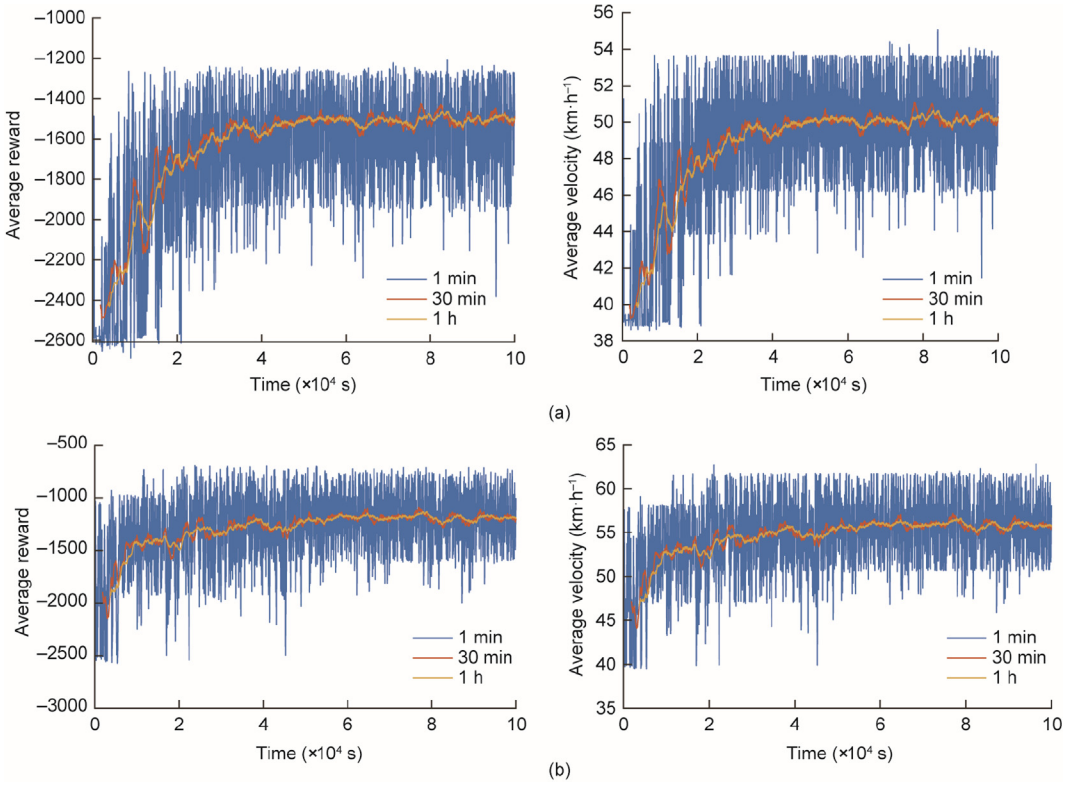

Fig. 11. Results of case 3: different traffic flow densities. (a) Longitudinal distance between traffic vehicles in the same lane: 80 m; (b) longitudinal distance between traffic vehicles in the same lane: 120 m.

traffic flow, as well as the planning layer style; ② the more aggressive the planning style is, the higher the utilization of the traffic flow free space will be, and the more possible it is to pursue a higher average speed by changing lanes frequently; and ③ sparse traffic flow allows the ego vehicle to evolve to provide more accelerating space, so that it can reach a higher average speed.

In our future work, we will focus on enabling the agent to learn the MPC parameters together with the proposed framework to improve the decision-making and motion planning flexibility; we will also investigate more driving tasks under this framework and conduct real vehicle experiments.

Acknowledgments

The authors would like to acknowledge the financial support of the National Key Research and Development Program of China (2020AAA0108100), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100), the Shanghai Gaofeng and Gaoyuan Project for University Academic Program Development for funding, and thank the anonymous reviewers for their valuable suggestions.

Compliance with ethics guidelines

Kang Yuan, Yanjun Huang, Shuo Yang, Zewei Zhou, Yulei Wang, Dongpu Cao, and Hong Chen declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Nilsson J, Brännström M, Coelingh E, Fredriksson J. Lane change maneuvers for automated vehicles. IEEE Trans Intell Transp Syst 2016;18(5):1087–96.
[2] Wang X, Qi X, Wang P, Yang J. Decision making framework for autonomous vehicles driving behavior in complex scenarios via hierarchical state machine. Auton Intell Syst 2021;1(1):1–12.
[3] Noh S. Decision-making framework for autonomous driving at road intersections: safeguarding against collision, overly conservative behavior, and violation vehicles. IEEE Trans Ind Electron 2018;66(4):3275–86.
[4] Nilsson J, Sjöberg J. Strategic decision making for automated driving on two-lane, one way roads using model predictive control. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV); 2013 Jun 23–26; Gold Coast, QLD, Australia. New York City: IEEE; 2013. p. 1253–8.
[5] Du Y, Wang Y, Chan CY. Autonomous lane-change controller via mixed logical dynamical. In: Proceedings of 17th International IEEE Conference on Intelligent Transportation Systems (ITSC); 2014 Oct 8–11; Qingdao, China. New York City: IEEE; 2014. p. 1154–9.
[6] Zhou Z, Yang Z, Zhang Y, Huang Y, Chen H, Yu Z. A comprehensive study of speed prediction in transportation system: from vehicle to traffic. iScience 2022;25(3):103909.
[7] Karlsson J, Murgovski N, Sjöberg J. Optimal trajectory planning and decision making in lane change maneuvers near a highway exit. In: Proceedings of 18th European Control Conference (ECC); 2019 Jun 25–28; Naples, Italy. New York City: IEEE; 2019. p. 3254–60.
[8] Nilsson J, Silvlin J, Brannstrom M, Coelingh E, Fredriksson J. If, when, and how to perform lane change maneuvers on highways. IEEE Intell Transp Syst Magazine 2016;8(4):68–78.
[9] Cui Z, Hu J, Guan H. A lane-changing trajectory planning and assistant decision-making method for autonomous vehicle. In: Proceedings of 18th COTA International Conference of Transportation Professionals (CICTP); 2018 Jul 5–8; Beijing, China. Reston: ASCE; 2018. p. 87–101.
[10] Xu D, Ding Z, He X, Zhao H, Moze M, Aioun F, et al. Learning from naturalistic driving data for human-like autonomous highway driving. IEEE Trans Intell Transp Syst 2020;22(12):7341–54.
[11] Liu Y, Zhou B, Wang X, Li L, Cheng S, Chen Z, et al. Dynamic lane-changing trajectory planning for autonomous vehicles based on discrete global trajectory. IEEE Trans Intell Transp Syst 2022;23(7):8513–27.
[12] Van Hoek R, Ploeg J, Nijmeijer H. Cooperative driving of automated vehicles using B-splines for trajectory planning. IEEE Trans Intell Vehicles 2021;6(3):594–604.
[13] Kim D, Jeong Y, Chung CC. Lateral vehicle trajectory planning using a model predictive control scheme for an automated perpendicular parking system. IEEE Trans Ind Electron 2023;70(2):1820–9.
[14] Mai TA, Dang TS, Duong DT, Le VC, Banerjee S. A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots. J Braz Soc Mech Sci Eng 2021;43(3):1–13.
[15] Moshayedi AJ, Li J, Liao L. Simulation study and PID tune of automated guided vehicles (AGV). In: Proceedings of IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement

Systems and Applications (CIVEMSA); 2021 Jun 18–20; Hong Kong, China. New York City: IEEE. p. 1–7.
[16] Sabiha AD, Kamel MA, Said E, Hussein WM. ROS-based trajectory tracking control for autonomous tracked vehicle using optimized backstepping and sliding mode control. Robot Auton Syst 2022;152:104058.
[17] El Atwi H, Daher N. A composite model predictive and super twisting sliding mode controller for stable and robust trajectory tracking of autonomous ground vehicles. In: Proceedings of IEEE 3rd International Multidisciplinary Conference on Engineering Technology (IMCET); 2021 Dec 8–10. Beirut, Lebanon. New York City: IEEE; 2022. p. 107–12.
[18] Ji J, Khajepour A, Melek WW, Huang Y. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. IEEE Trans Vehicular Technol 2016;66(2):952–64.
[19] Huang Y, Wang H, Khajepour A, Ding H, Yuan K, Qin Y. A novel local motion planning framework for autonomous vehicles based on resistance network and model predictive control. IEEE Trans Vehicular Technol 2019;69 (1):55–66.
[20] Wischnewski A, Herrmann T, Werner F, Lohmann B. A tube-MPC approach to autonomous multi-vehicle racing on high-speed ovals. IEEE Trans Intell Vehicles 2023;8(1):368–78.
[21] Evens B, Schuurmans M, Patrinos P. Learning MPC for interaction-aware autonomous driving: a game-theoretic approach. 2021. arXiv: 2111.08331.
[22] Yuan K, Shu H, Huang Y, Zhang Y, Khajepour A, Zhang L. Mixed local motion planning and tracking control framework for autonomous vehicles based on model predictive control. IET Intell Transp Syst 2019;13(6):950–9.
[23] Mohseni F, Frisk E, Nielsen L. Distributed cooperative MPC for autonomous driving in different traffic scenarios. IEEE Trans Intell Vehicles 2020;6(2):299–309.
[24] Huang Y, Ding H, Zhang Y, Wang H, Cao D, Xu N, et al. A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach. IEEE Trans Ind Electron 2020;67 (2):1376–86.
[25] Zhou Q, Zhao D, Shuai B, Li Y, Williams H, Xu H. Knowledge implementation and transfer with an adaptive learning network for real-time power management of the plug-in hybrid vehicle. IEEE Trans Neural Netw Learn Syst 2021;32(12):5298–308.
[26] Wang Y, Zhang D, Wang J, Chen Z, Li Y, Wang Y, et al. Imitation learning of hierarchical driving model: from continuous intention to continuous trajectory. IEEE Robot Autom Lett 2021;6(2):2477–84.
[27] Hoel CJ, Wolff K, Laine L. Automated speed and lane change decision making using deep reinforcement learning. In: Proceedings of 21st International Conference on Intelligent Transportation Systems (ITSC); 2018 Nov 4–7; Maui, HI, USA. New York City: IEEE; 2018. p. 2148–55.
[28] Liu Y, Wang X, Li L, Cheng S, Chen Z. A novel lane change decision-making model of autonomous vehicle based on support vector machine. IEEE Access 2019;7:26543–50.
[29] Wang X, Wu J, Gu Y, Sun H, Xu L, Kamijo S, et al. Human-like maneuver decision using LSTM-CRF model for on-road self-driving. In: Proceedings of 21st International Conference on Intelligent Transportation Systems (ITSC); 2018 Nov 4–7; Maui, HI, USA. New York: IEEE; 2018. p. 210–6.
[30] Xiao Y, Codevilla F, Gurram A, Urfalioglu O, L'opez AM. Multimodal end-to-end autonomous driving. IEEE Trans Intell Transp Syst 2020;23(1):537–47.
[31] Menner M, Berntorp K, Zeilinger MN, Di Cairano S. Inverse learning for data-driven calibration of model-based statistical path planning. IEEE Trans Intell Vehicles 2020;6(1):131–45.
[32] Peng B, Sun Q, Li SE, Kum D, Yin Y, Wei J, et al. End-to-end autonomous driving through dueling double deep Q-network. Automotive Innovation 2021;4 (3):328–37.
[33] Lin Y, McPhee J, Azad NL. Comparison of deep reinforcement learning and model predictive control for adaptive cruise control. IEEE Trans Intell Vehicles 2020;6(2):221–31.
[34] He X, Yang H, Hu Z, Lv C. Robust lane change decision making for autonomous vehicles: an observation adversarial reinforcement learning approach. IEEE Trans Intell Vehicles 2023;8(1):184–93.
[35] Li G, Li S, Li S, Qin Y, Cao D, Qu X, et al. Deep reinforcement learning enabled decision-making for autonomous driving at intersections. Automotive Innovation 2020;3(4):374–85.
[36] Liu Z, Hu J, Song T, Huang Z. A methodology based on deep reinforcement learning to autonomous driving with double Q-Learning. In: Proceedings of 7th International Conference on Computer and Communications (ICCC); 2021 Dec 10–13; Chengdu, China. New York City: IEEE; 2022. p. 1266–71.
[37] Aradi S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Trans Intell Transp Syst 2022;23(2):740–59.
[38] Li G, Yang Y, Li S, Qu X, Lyu N, Li SE. Decision making of autonomous vehicles in lane change scenarios: deep reinforcement learning approaches with risk awareness. Transp Res Part C 2022;134:103452.
[39] Zhang Y, Sun P, Yin Y, Lin L, Wang X. Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV); 2018 Jun 26–30; Changshu, China. Changshu, China. New York City: IEEE; 2018. p. 1251–6.