J Intel Connect Veh

Research Article

Prior-knowledge-guided model-based reinforcement learning for integrated longitudinal-lateral control of vehicular platoons

Xia Wu[1], Haigen Min[1,✉], Zihao Mao[1], Yanbing Yan[1], Yang Liu[2], Guoyuan Wu[3]

Cite this article: Wu X, Min H G, Mao Z H, et al. *J Intel Connect Veh* 2026, **9**(2): 9210083. https://doi.org/10.26599/JICV.2026.9210083

ABSTRACT: To address the complexity of longitudinal-lateral vehicle dynamics in platoon control and the challenge of mitigating traffic oscillations, a novel model-based reinforcement learning (MBRL) method with planning capability is proposed for longitudinal-lateral control of vehicle platoons. Specifically, the integration of the environment model and policy model, combined with a sampling-based planning approach, enables planning-based control during decision-making, which significantly improves the stability and safety of vehicle platoon control. In the two-dimensional scenario, a unified state space and joint action space are designed to achieve longitudinal-lateral control, along with a multidimensional reward function that integrates both control objectives. To address the low exploration efficiency and the high proportion of ineffective exploration during the early stage of training, a prior-knowledge-guided exploration strategy is introduced. This strategy improves learning efficiency and accelerates convergence by incorporating guided actions and constraints. The training results indicate that the incorporation of prior knowledge significantly enhances training efficiency. In evaluations under speed-limit scenarios and real-data scenarios, the proposed method demonstrates superior performance in longitudinal-lateral control, platoon stability, safety, and adaptability, while maintaining high computational efficiency.

KEYWORDS: deep reinforcement learning; model-based reinforcement learning (MBRL); prior knowledge; vehicle platoon

1 Introduction

Connected and automated vehicles (CAVs) are widely regarded as a key development direction for future highway and urban road transportation. Integrating advanced sensing, automation, computing, and communication technologies, CAVs demonstrate enormous potential in enhancing traffic safety and operational efficiency (Elliott et al., 2019; Guanetti et al., 2018). Leveraging cutting-edge control methods, CAVs can optimize driving behaviors and improve the overall traffic flow capacity. In particular, CAVs can dynamically adjust their driving decisions by utilizing environmental perception information (e.g., the status of surrounding vehicles and road geometry features), enabling more efficient cooperative driving. Through vehicle-to-vehicle (V2V) communication, CAVs can form platoons that enhance road capacity and driving safety while effectively reducing vehicle energy consumption and mitigating "stop-and-go" traffic oscillations (Li et al., 2010). However, ensuring a stable platoon operation and optimizing overall traffic flow remain significant challenges, especially in two-dimensional geometric environments. Designing effective platooning control methods is therefore a core issue for achieving safety and stability.

In recent years, research on vehicle platooning control has advanced significantly, with mainstream control methods categorized into three types: (1) analytical linear or nonlinear methods, (2) model predictive control (MPC)-based methods, and (3) deep reinforcement learning (DRL)-based methods. Analytical linear or nonlinear methods (Su et al., 2024; Wang et al., 2020; Zhu et al., 2022) offer high computational speed, ease of implementation, and support stability analysis with local stability and string stability verifiable via mathematical theories. However, these methods face difficulties in addressing multiobjective optimization, complex constraints, and nonlinear environments, restricting their practical applications. MPC-based methods (Pi et al., 2022; Wang et al., 2022; Yan et al., 2019) provide a more flexible optimization framework, enabling the optimization of multiple control objectives and consideration of constraints within a rolling time horizon. Nevertheless, MPC typically requires the optimization problem to be convex and involves high computational complexity, presenting challenges for high-dynamic environments or applications with strict real-time requirements.

DRL-based methods (Jiang et al., 2022; Li et al., 2021; Qu et al., 2020; Shi et al., 2022; Yan et al., 2022; Yu et al., 2024) train policy networks offline and leverage neural networks to learn control strategies for complex systems, effectively capturing the systems' stochastic characteristics. Since the main computational burden is concentrated in the training phase, DRL enables fast computations and real-time control during inference. However, most existing DRL-based methods rely on model-free reinforcement learning (MFRL), where decisions are solely based on the trained policy network. This can lead to poor performance

[1] School of Information Engineering, Chang'an University, Xi'an 710064, China. [2] School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore. [3] School of Electrical and Computer Engineering, University of California, California 92521, USA.

✉ Corresponding author. E-mail: hgmin@chd.edu.cn

Received: January 14, 2026; Revised: March 18, 2026; Accepted: May 4, 2026

in unseen scenarios and issues such as low learning efficiency and lack of interpretability (Pal and Leon, 2020). In contrast, model-based reinforcement learning (MBRL) methods learn the policy while simultaneously acquiring the environment's internal model. By constructing and leveraging this model, the agent can enhance learning efficiency without requiring frequent interactions with the real-world environment. Yavas et al. (2022) proposed an MBRL-based adaptive cruise control method that uses the learned environment model to predict the internal dynamics of the traffic environment, avoiding passive short-term strategies. The results demonstrate that this method achieves superior control performance and significantly enhanced sample efficiency. However, the employed MBRL method did not fully exploit the potential of the learned environment model, using it merely to reduce interactions with the real-world environment (Dong et al., 2024). Indeed, the model can also support trajectory predictive planning during decision-making, facilitating better action selection and improved system control performance (Kotb et al., 2023), thereby further enhancing overall performance and safety.

Most existing studies focus on longitudinal control of vehicle platooning. However, in real-world scenarios, roads are not always straight, requiring simultaneous consideration of both longitudinal and lateral platoon control. While some studies have started to explore the coupling of lateral and longitudinal control, most methods still treat them separately, failing to fully capture the inherent coupling between the two (Zhang et al., 2022). Furthermore, these methods often assume homogeneous road geometric features, such as constant curvature (Fujimoto et al., 2018), an assumption that restricts the adaptability of control methods in complex traffic environments. For example, Zhang et al. (2022) proposed a cooperative adaptive cruise control method based on MPC that separately controls longitudinal and lateral motions. Longitudinal control used MPC to gradually converge the following distance to the desired value while ensuring that the speed and acceleration were as close as possible to the lead vehicle. Lateral control was also based on MPC, minimizing the lateral offset, angle error, and steering angle. However, this method does not consider the coupling between longitudinal and lateral dynamics, and due to the complexity of lateral state transitions, it uses an approximation that does not fully account for the impact of road curvature. Ren et al. (2023) proposed an adaptive cooperative cruise control algorithm based on the Frenet coordinate system, decoupling the vehicle's two-dimensional complex motion into two one-dimensional motions and designing separate longitudinal and lateral control strategies. This simplified the controller design and improved the computational efficiency. Nevertheless, this method also failed to fully consider the mutual influence between longitudinal and lateral dynamics and assumed constant road curvature.

Vehicle longitudinal and lateral dynamics are highly interconnected and exhibit time-varying nonlinear characteristics (Li et al., 2022). For example, fluctuations in longitudinal speed can lead to lateral offset, which in turn affects the stability of path tracking. Therefore, optimizing the longitudinal and lateral control in a separate manner (for the sake of system design simplification) compromises the overall system performance (Shan et al., 2024). Meanwhile, road curvature often varies in real-world situations, and control strategies designed under the fixed curvature assumption fail to meet the vehicle's demand for high-precision and robust control in complex road scenarios (Barreno et al., 2025). Thus, designing control methods that simultaneously consider vehicle longitudinal and lateral control while integrating dynamic road geometric features can enhance adaptability and improve control performance.

Reinforcement learning (RL) is an interaction-driven learning paradigm in which an agent interacts with the environment to collect experience data and learn the optimal policy (Arulkumaran et al., 2017). In model-free reinforcement learning, the agent relies solely on environmental feedback to optimize the policy, whereas MBRL learns the environment's dynamic model alongside the policy, enabling future state prediction (Chatterjee and Khardon, 2025). Furthermore, MBRL relies on temporal data for training to ensure that the model can accurately simulate environmental evolution. However, the environment may contain dangerous states; once the agent enters these states, training may be interrupted, compromising data collection continuity (Liu and Wang, 2021). In RL, the most common exploration strategy is random exploration, where noise is randomly added to any given state to enhance policy diversity. However, random exploration has significant limitations, especially in the early stages of training. The agent often performs ineffective actions or even enters dangerous states, leading to premature episode termination. This results in insufficient data collection, which ultimately hinders training effectiveness and slows convergence speed (Liu et al., 2020). These issues can often be mitigated by incorporating domain-specific prior knowledge of the target environment. When appropriately integrated into the RL training process, this prior knowledge helps the agent acquire superior initial strategies, thereby improving early-stage performance, reducing ineffective exploration, enhancing the quality and efficiency of training samples, and ultimately enabling safer and more efficient training.

To address the aforementioned challenges, this study innovatively proposes a prior-knowledge-integrated model-based reinforcement learning (MBRL) method with planning capabilities for vehicle platoon longitudinal-lateral control. This approach ensures lateral control accuracy, longitudinal control stability, and effective suppression of traffic oscillations. The main contributions of the study are summarized as follows:

A unified state space, joint action space, and integrated reward function are constructed under the Frenet coordinate system. The unified state space integrates vehicle and environmental states to comprehensively capture dynamic characteristics; the joint action space models lateral and longitudinal control inputs in a unified manner; and the integrated reward function comprehensively considers longitudinal-lateral control performance, riding comfort, and safety.

A planning-enabled vehicle platoon control method is proposed. By fusing the environment model and policy model from MBRL, a sampling-based planning mechanism is introduced into the decision-making process. Local evaluation is performed via the reward model, and global trajectory evaluation is conducted using the state value function. The action sequence is optimized based on these evaluations, with the first action of the final sequence executed at each decision step.

A prior-knowledge-incorporated exploration strategy is proposed to address low exploration efficiency and a high proportion of ineffective exploration in the early training stage under a two-dimensional action space. Expert strategies derived from prior knowledge are leveraged to guide the initial exploration direction, and action constraints are established to reduce ineffective trials, thereby achieving faster convergence and higher learning efficiency.

2 Problem formulation

2.1 Vehicle kinematic model

The vehicle platoon environment is considered on a single-lane

road with varying road geometries (i.e., roads with different curvatures). The CAVs are equipped with V2V communication technologies to collect real-time information about the surrounding environment, including nearby vehicles and road conditions for decision-making. The basic assumptions of the simulation environment are as follows: (1) V2V communication topology adopts a widely used broadcast mechanism, enabling V2V to broadcast their state information (e.g., position and speed); (2) CAVs can receive real-time information about local road attributes (e.g., road curvature and speed limits) by communicating with roadside units; (3) due to the increasing maturity of advanced communication technologies, communication delays and packet loss are not considered; and (4) lane changes are not considered within the vehicle platoon.

In the environment, vehicles are modeled using the widely adopted kinematic bicycle model, which takes the rear axle as the reference point. In this model, the four wheels are simplified into a pair of wheels located at the center points of the front and rear axles. Only the front wheel is steerable, thereby controlling the vehicle's lateral motion. Let x_i^t and y_i^t denote the vehicle's lateral and longitudinal positions, respectively, and θ_i^t denote its heading angle. The vehicle's state in the Cartesian coordinate system can be represented as a vector $[x_i^t, y_i^t, \theta_i^t]$. The kinematics of the vehicle can then be described as Eqs. (1)–(4):

$$\dot{x}_i^t = v_i^t \cos\theta_i^t \tag{1}$$

$$\dot{y}_i^t = v_i^t \sin\theta_i^t \tag{2}$$

$$\theta_i^t = v_i^t \tan(\delta_i^t)/L \tag{3}$$

$$\dot{v}_i^t = a_i^t \tag{4}$$

where v_i^t, a_i^t, and δ_i^t represent the velocity, acceleration, and steering angle of vehicle i at time t, respectively. The parameter L denotes the wheelbase of the vehicle, which is the distance between the front and rear axles.

Since the vehicle's acceleration and steering angle cannot change instantaneously, actuator delay is also considered in the construction of the two-dimensional environment. Actuator delay refers to the time lag between issuing a control command and the actual execution of that command. It is approximated using a first-order time delay process. The actuator delays are given as Eq. (5) and (6):

$$\dot{a} = \frac{u_a - a}{\tau_a} \tag{5}$$

$$\dot{\delta} = \frac{u_\delta - \delta}{\tau_\delta} \tag{6}$$

where u_a and u_δ denote the desired acceleration and steering angle, respectively, while a and δ represent the actual acceleration and steering angle, respectively. The parameters τ_a and τ_δ are the time constants for the longitudinal and lateral control delays, respectively.

Note that the control action produced by the agent is the desired command $u_t = [a_d, \delta_d]$. The simulator updates the actual actuator states (a_t, δ_t) through the first-order delay dynamics in Eqs. (5) and (6), and these actuator states are included in the unified state vector in Eq. (14). Therefore, the transition samples used to train the learned dynamics model are generated under the same actuator delay, so the learned environment model implicitly captures the delay behavior. Any remaining model mismatch is further mitigated by receding-horizon replanning and by keeping the rollout horizon short to reduce error accumulation.

The reference path is curved, resulting in a nonlinear vehicle-following process that becomes complicated when described in the Cartesian coordinate system (Moghadam et al., 2021). Therefore, the Frenet coordinate system is adopted to simplify the description of vehicle motion. The vehicle state can be represented as a three-dimensional vector $[p_i^t, e_i^t, \varphi_i^t]$. p_i^t denotes the longitudinal projection of vehicle i at time t, which is the route length from the starting point of the road to projection point M on the reference path. e_i^t represents the lateral deviation from the reference path, and φ_i^t denotes the heading angle error between the vehicle and the tangent of the reference path at point M. The transformation between the Cartesian and Frenet coordinate systems for the bicycle model is illustrated in Fig. 1.

The lateral deviation e_i^t of vehicle i from the reference trajectory at time t is defined as the distance between the rear axle of the vehicle and the projection point $M(x_{\text{ref}}, y_{\text{ref}})$ on the reference path. It is mathematically computed as Eq. (7):

$$\begin{cases} e_i^t = b \times d \\ b = \text{sign}\left[\left(y_i^t - y_{i,\text{ref}}^t\right)\cos\left(\theta_{\text{ref}}^t\right) - \left(x_i^t - x_{i,\text{ref}}^t\right)\sin\left(\theta_{i,\text{ref}}^t\right)\right] \\ d = \sqrt{\left(x_i^t - x_{i,\text{ref}}^t\right)^2 + \left(y_i^t - y_{i,\text{ref}}^t\right)^2} \end{cases} \tag{7}$$

where x_{ref}, y_{ref}, and θ_{ref} represent the x-coordinate, y-coordinate, and heading angle of the projection point M on the reference path, respectively. The first component b indicates the direction of the lateral deviation. It is positive when the vehicle is to the left of the reference path (relative to the direction of travel) and negative when to the right. The second component d represents the distance from the vehicle to the projection point M.

The heading angle error φ_i^t of vehicle i with respect to the reference trajectory at time t is defined as the difference between the vehicle's heading angle and the heading angle of the reference path at point M. It is given by

$$\varphi_i^t = \theta_i^t - \theta_{i,\text{ref}}^t \tag{8}$$

The longitudinal projection velocity q_i^t of vehicle i along the reference path at time t is given by

$$q_i^t = \frac{v_i^t \cos(\varphi_i^t)}{1 - e_i^t k_i^t} \tag{9}$$

where k_i^t denotes the road curvature at point M on the reference path for vehicle i at time t.

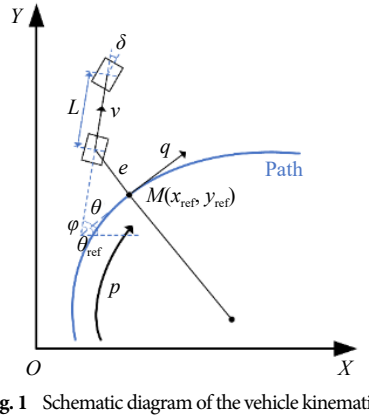


Fig. 1 Schematic diagram of the vehicle kinematics.

2.2 Vehicle platoon design

A homogeneous platoon consisting of one lead vehicle and $n-1$ (2D) follower vehicles is considered, traveling on a two-dimensional, long, and flat road, as illustrated in Fig. 2. The desired vehicle spacing follows a constant time gap policy. Due to the curved motion in the two-dimensional plane, the vehicle speed is represented by the longitudinal projection velocity. Accordingly, the desired spacing between vehicle i and its preceding vehicle at time t is given by

$$d_{i,i-1}^{*,t} = q_i^t \varepsilon + l \tag{10}$$

where l denotes the standstill spacing between vehicle i and its predecessor, and ε represents the desired time gap.

To perform longitudinal and lateral control of vehicles in the Frenet coordinate system, it is necessary to obtain the longitudinal projection spacing error Δs_i^t, the longitudinal projection velocity error Δq_i^t, the lateral deviation e_i^t, and the heading angle deviation φ_i^t.

The longitudinal projection space error between vehicle i and its preceding vehicle at time t, denoted as Δs_i^t, is defined as the difference between the actual intervehicle spacing and the desired spacing, as given below:

$$\Delta s_{i,i-1}^t = d_{i,i-1}^t - d_{i,i-1}^{*,t} \tag{11}$$

$$d_{i,i-1}^t = p_{i-1}^t - p_i^t - c \tag{12}$$

where $d_{i,i-1}^t$ denotes the spacing between vehicle i and its preceding vehicle, and c denotes the length of the vehicle.

The longitudinal projection velocity error, denoted as Δq_i^t, is defined as the difference between the current longitudinal projection velocity of vehicle i and the desired projection velocity.

The desired projection velocity and the velocity error are given as Eq. (13):

$$\Delta q_{i,i-1}^t = q_i^t - q_{i-1}^{*,t} = q_i^t - q_{i-1}^t \tag{13}$$

2.3 Road and lead vehicle strategy design

To realize longitudinal and lateral control of vehicle platoons in a two-dimensional environment, an accurate and smooth 2D reference road trajectory is essential. Therefore, the environment must possess an efficient path generation capability to provide the necessary trajectory information for platoon control. A two-dimensional cubic spline interpolation method is employed to generate a continuous and smooth reference road trajectory based on a set of discrete road center points.

It is worth noting that to comply with road geometric design standards, the generated path curvature must be kept within a reasonable range. In this study, a maximum curvature constraint of $k < 0.25$ is imposed to avoid sharp turns that could lead to control difficulties and safety risks.

In platoon control, the lead vehicle strategy refers to the control behavior of the first vehicle in the platoon. This ensures the proper operation of the entire platoon, enabling the follower vehicles to travel stably along the reference road. The lead vehicle strategy generates control inputs such as acceleration and steering angle based on current speed limits, road curvature, and its own state. A proportional integral derivative (PID) controller is applied for longitudinal control, adjusting acceleration to bring the vehicle speed closer to the desired value and ensure smooth driving. At the same time, a linear quadratic regulator (LQR) is used for lateral control, optimizing the steering angle to enable accurate tracking of the target path.

3 Methodology

3.1 Method framework

This section proposes a model-based reinforcement learning (MBRL) framework with planning capabilities for vehicle platoon control, which explicitly integrates prior knowledge, as illustrated in Fig. 3. The framework operates as a closed-loop control system comprising four synergistic components: the vehicle platoon environment, a planning-based agent, probabilistic network models, and an experience replay buffer. The environment simulates the coupled longitudinal and lateral dynamics of the platoon under complex road geometries. At each time step, it provides the agent with a state vector containing vehicle kinematics and tracking errors while simultaneously computing a multiobjective reward function that balances safety, comfort, and control efficiency to guide the optimization process. Central to this framework is the agent, which advances beyond simple reactive policies by employing a model-based trajectory planning mechanism. Specifically, the agent utilizes the actor network to generate a distribution of candidate action sequences. These candidates are rigorously refined by a prior knowledge module, which acts as a filter to prune physically infeasible or unsafe actions. The agent then leverages the learned probabilistic environment model—comprising transition and reward models—to predict the future trajectories of the remaining candidates. By evaluating the expected long-term returns of these trajectories, the agent identifies and executes the optimal "elite action sequence". To ensure continuous improvement, all interaction transitions are stored in the experience replay buffer

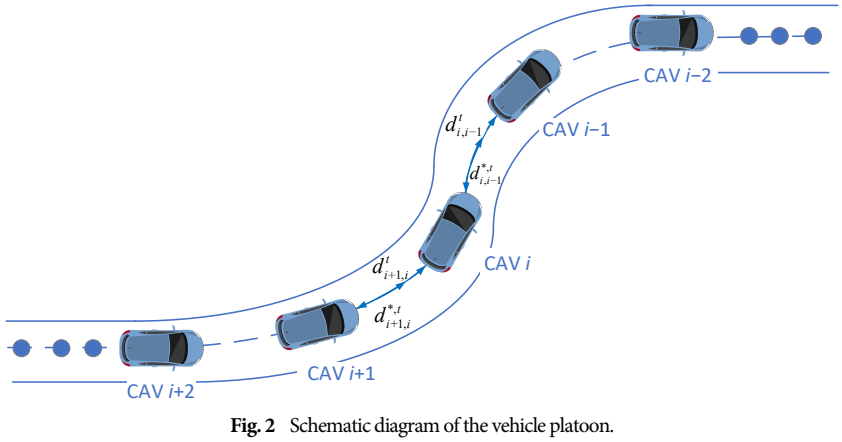


Fig. 2 Schematic diagram of the vehicle platoon.

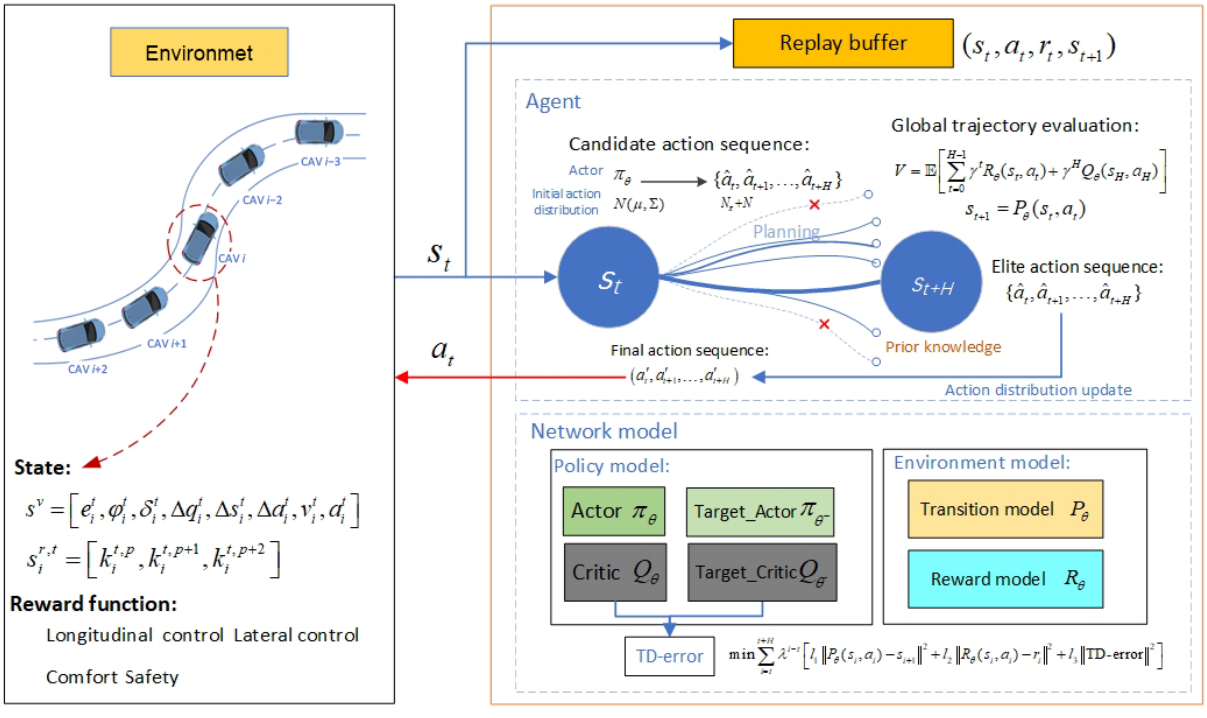


Fig. 3 Framework of the proposed planning-based MBRL method integrating prior knowledge for vehicle platoon control.

and utilized for off-policy training, alternately updating the environment model and policy networks to enhance both prediction accuracy and control robustness.

Specifically, the environment is established based on the Frenet coordinate system introduced in the previous chapter to facilitate both longitudinal and lateral control. The reward function is designed to comprehensively reflect the effectiveness of tracking control as well as driving comfort and safety. An experience replay buffer is employed to store the sequential data (state, action, reward, and next state) generated during the interaction between the agent and the environment. These samples are then used to train the models, enabling the agent to make more informed decisions and enhanced its learning process.

The network architecture consists of two main components: the policy model and the environment model. The policy model is composed of an actor network, a critic network, and their corresponding target networks. The environment model comprises a state transition model and a reward model.

Finally, during the planning phase, the agent utilizes both the policy and environment models. The actor network, together with an initial action distribution, generates a set of candidate action sequences. These sequences are then globally evaluated, and the elite action sequences with the best performance are selected to update the optimal action distribution. This process is iteratively refined to obtain the final action sequence for execution.

3.2 Reinforcement learning model

Reinforcement learning is a method in which an agent learns how to select actions through interactions with the environment to maximize its long-term returns. The core idea lies in the agent's ability to interact with the environment through states, receive feedback in the form of rewards based on its actions, and adjust its behavior accordingly to achieve task objectives. In an RL framework, the state space, action space, and reward function constitute the key components that define the learning process.

The state space represents the information that the agent can perceive about itself and its surrounding environment. In the current longitudinal-lateral vehicle platooning control problem, a unified state space is constructed by integrating both the vehicle's state and the environmental state. It is assumed that the preceding vehicle and the ego vehicle can communicate their motion information in real time via onboard communication devices. The complete state space is denoted as $[s^v, s^e]$, where s^v represents the vehicle's state and s^e denotes the road state. The vehicle state includes both lateral and longitudinal states.

The vehicle's lateral state includes the lateral distance deviation, heading angle deviation, and steering angle. The longitudinal state includes the longitudinal projection spacing error, longitudinal projection velocity error, velocity, and acceleration. The complete vehicle state is represented as Eq. (14):

$$s^v = \left[e_i^t, \varphi_i^t, \delta_i^t, \Delta s_i^t, \Delta q_i^t, v_i^t, a_i^t\right] \tag{14}$$

The road state mainly considers the curvature of the road, including the road curvature at the vehicle's projection point at the current time, as well as the curvature 1 and 2 m ahead of the projection point, as shown below:

$$s^e = [k_i^{t,p}, k_i^{t,p+1}, k_i^{t,p+2}] \tag{15}$$

In the study of longitudinal-lateral control for vehicle platoons, a joint action space is designed to further achieve coordinated optimization of both control dimensions. By constructing a combined action space $u = [u_a, u_\delta]$ that includes both the desired acceleration and steering angle, the agent can consider the effects of both control aspects simultaneously during learning, effectively avoiding the information fragmentation caused by separated control strategies. The control input is constrained within a specified range, denoted as $-u_a^{\min} < u_a < u_a^{\max}$, $-u_\delta^{\min} < u_\delta < u_\delta^{\max}$.

Although longitudinal car-following and lateral path tracking can be treated separately in simplified designs, platoon dynamics in the Frenet frame are inherently coupled. First, the longitudinal projection speed depends on both heading error and lateral deviation through Eq. (9), where the term $(1 - ke_y)$ couples the

lateral position to longitudinal progress. Second, steering actions affect e_y and e_ψ, which then influence the longitudinal spacing error through s, while acceleration changes v and thus impacts both lateral motion and comfort-related quantities. By including both longitudinal errors (Δs, Δv, v, a) and lateral errors (e_y, e_ψ, δ), together with road curvature, in one state vector and optimizing the joint action (a_d, δ_d) under a single reward, the learned transition model and the MPPI planner can explicitly reason about these cross-effects, unlike traditional decoupled controllers that solve two separate one-dimensional subproblems.

The reward function serves as guidance for the agent to learn optimal policies by maximizing the cumulative reward. To enable coordinated optimization of both lateral and longitudinal control strategies in vehicle platoon control, a composite reward function is designed. This function comprehensively considers lateral control performance, longitudinal control performance, comfort, and safety.

The first component of the reward focuses on longitudinal control performance, which includes the longitudinal projection distance error and the longitudinal projection velocity error, as shown below:

$$r_1 = (\Delta s_i)^2 \tag{16}$$

$$r_2 = (\Delta q_i)^2 \tag{17}$$

The second component addresses the lateral control performance of the vehicle, which includes the lateral distance deviation and the heading angle deviation, as formulated below:

$$r_3 = (e_i)^2 \tag{18}$$

$$r_4 = (\varphi_i)^2 \tag{19}$$

The third component is designed to optimize driving comfort. Previous studies often penalized the magnitude of control inputs; however, this approach limits the vehicle's dynamic capabilities without directly addressing passenger discomfort. To effectively improve ride quality, we focus on minimizing the rate of change of the vehicle's dynamics. Specifically, the reward function penalizes longitudinal jerk (the derivative of acceleration) and the steering rate (the derivative of the steering angle). Additionally, excessive lateral acceleration is penalized to prevent discomfort during cornering. This formulation ensures smooth transitions in vehicle states, thereby enhancing stability and passenger comfort, as defined below:

$$r_5 = (u_{a,t} - u_{a,t-1})^2 + (u_{\delta,t} - u_{\delta,t-1})^2 \tag{20}$$

The fourth factor addresses safety by imposing a penalty when a collision occurs between the ego vehicle and the vehicle ahead. Incorporating such collision feedback during training enables the agent to learn to avoid dangerous situations. Specifically, a collision is defined when the relative distance between the two vehicles becomes negative. Unlike a fixed penalty, the safety reward is designed as a continuous, distance-dependent signal. This formulation provides the reinforcement learning agent with a meaningful gradient even inside the collision state, encouraging it to minimize both the depth of overlap and the magnitude of the spacing error. As a result, the policy is guided to actively recover from collisions rather than stagnating in a state of persistent penalty, thereby avoiding convergence to a suboptimal local solution. When no collision occurs, this reward component remains zero.

$$r_6 = \begin{cases} -10 + 5(\Delta s_i), d_{i,i-1} < 0 \\ 0, \text{else} \end{cases} \tag{21}$$

The complete reward function integrates considerations of lateral control performance, longitudinal control performance, comfort, and safety, as shown below:

$$r = k_1 r_1 + k_2 r_2 + k_3 r_3 + k_4 r_4 + k_5 r_5 + k_6 r_6 \tag{22}$$

where k_1, k_2, k_3, k_4, k_5, and k_6 denote the weights assigned to different components of the reward function. These weights were determined through a two-phase procedure. First, initial values were selected to normalize the different physical scales of the reward terms. Subsequently, the weights were systematically fine-tuned via grid search, prioritizing safety and stability over comfort. This tuning process aimed to maximize the primary task performance while strictly satisfying the constraints on jerk and safety. After conducting numerous experiments, the final weight values adopted were −8, −4, −6, −1, −1, and 1.

3.3 Planning-capable MBRL approach

Model-based reinforcement learning methods with planning capabilities offer significant advantages in terms of sample efficiency, asymptotic performance, and interpretability. By leveraging learned models for planning, decision-making can be enabled through both environmental and policy models. The environment model refers to the internal modeling of the environment, including a state transition model and a reward model. The learned state transition model allows the agent to predict the next state based on the current state and action, while the learned reward model enables the agent to estimate the current reward. With these two models, the agent can simulate outcomes without interacting with the real environment. The formulations of the two models are as Eqs. (23) and (24):

$$\hat{s}_{t+1} = P_\theta(s_t, a_t) \tag{23}$$

$$\hat{r}_t = R_\theta(s_t, a_t) \tag{24}$$

Actuator delay handling in the learned environment model. The learned transition model P_θ is trained to predict s_{t+1} from (s_t, u_t), where $u_t = [a_d, \delta_d]$ is the desired command. Because the simulator updates the actual acceleration and steering states via Eqs. (5) and (6) and these states are part of s_t, the learned model captures the delayed actuator response implicitly from data. In addition, the MPPI planner uses frequent replanning (receding horizon) to correct for disturbances and residual prediction errors.

The policy model consists of an actor network and a critic network. The actor network generates actions based on the current state, which, when combined with the state transition model, can be used to generate candidate action sequences. The critic network evaluates the outcomes of these action sequences and provides feedback to improve the actor network. Specifically, the structures of the actor and critic networks are defined as Eqs. (25) and (26):

$$\hat{a}_t = \pi_\theta(s_t) \tag{25}$$

$$\hat{q}_t = Q_\theta(s_t, a_t) \tag{26}$$

The planning process adopts the Model Predictive Path Integral (MPPI) method, with the detailed steps as follows: (1) Use the policy network π and add noise to generate N_π candidate action

sequences. (2) Based on the initial action distribution $N(u, \Sigma)$, generate and construct N candidate action sequences. (3) Evaluate all $N + N_\pi$ candidate action sequences, perform global evaluation, and select the top N' sequences as elite action sequences. (4) Calculate the action distribution of the elite sequences and update the action distribution accordingly. This process is repeated for m iterations to ensure continuous improvement of the action distribution quality. Finally, an optimized action distribution is obtained. The final action sequence $(a_t^{'}, a_{t+1}^{'}, \ldots, a_{t+H}^{'})$ is taken as the mean of the updated distribution. The first action of this sequence is then executed in the environment, while the remaining actions are retained for generating the next final action sequence.

In each iteration, it first performs a global evaluation of all candidate action sequences. This evaluation method combines both local evaluation and terminal evaluation. The local evaluation uses the learned state transition model P_θ to simulate state transitions, followed by a discounted evaluation using the learned reward model R_θ. The terminal global evaluation is conducted using the Critic network Q_θ. The evaluation process is formulated as Eq. (27):

$$V = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t R_\theta(s_t, a_t) + \gamma^H Q_\theta(s_H, a_H)\right], s_{t+1} = P_\theta(s_t, a_t) \tag{27}$$

where γ is a discount factor for evaluation, used to adjust the importance of future evaluations.

All candidate action sequences are ranked based on their evaluation scores, and the top N' sequences are selected as elite action sequences ξ^*. The action distribution of these elite sequences is then computed as Eq. (28):

$$\mu^{'} = \frac{\sum_{j=1}^{N^{'}} e^{V_j^*} \xi_j^*}{\sum_{j=1}^{N^{'}} e^{V_i^*}}, \Sigma^{'} = \sqrt{\frac{\sum_{j=1}^{N^{'}} e^{V_j^*} \left(\xi_j^* - \mu^{'}\right)^2}{\sum_{j=1}^{N^{'}} e^{V_j^*}}} \tag{28}$$

where $e^{V_i^*}$ denotes the exponentially weighted value assigned to the elite action sequence.

At each time step, multiple iterations are performed through resampling and noise distribution updates. The action distribution is updated using a soft update strategy, as shown below:

$$\mu = (1-\sigma)\mu + \sigma\mu^{'}, \Sigma = (1-\sigma)\Sigma + \sigma\Sigma^{'} \tag{29}$$

where σ is the soft update coefficient for the action distribution, and $\sigma \in [0, 1]$.

The models to be learned include the state transition model P and reward model R of the environment, as well as the actor network π and critic network Q of the policy. These models are trained using sequential data collected in each episode and stored in the experience replay buffer. Specifically, during training, H-step trajectories $\Gamma(s_t, a_t, r_t, s_{t+1})$ are sampled from the buffer, and the models are updated by minimizing a loss function that incorporates H-step temporal discounting, as formulated below:

$$J(\theta; \Gamma) = \sum_{i=t}^{t+H} \lambda^{i-t} L(\theta; \Gamma_i) \tag{30}$$

where λ is the discount factor of the loss function, which assigns higher weights to near-term predictions. $L(\theta; \Gamma_i)$ denotes the objective function for trajectory Γ_i, which incorporates the state transition model, reward model, critic network, and target critic network, as shown below:

$$\begin{aligned} L(\theta; \Gamma_i) = {} & l_1 ||P_\theta(s_i, a_i) - s_{t+1}||^2 + l_2 ||R_\theta(s_i, a_i) - r_t||^2 \\ & + l_3 ||\text{TD\_error}||^2 \end{aligned} \tag{31}$$

where l_1, l_2, and l_3 are the loss balancing coefficients, and TD-error is defined as $Q_\theta(s_i, a_i) - (r_i + \gamma Q_\theta - (s_{i+1}, \pi_\theta(s_{i+1}))$. The target Critic network shares the same architecture as the Critic network and is updated using a soft update strategy.

The learning process of the policy network is similar to that in the DDPG (Lillicrap et al., 2019) algorithm used in MFRL. The actor network is trained by maximizing the discounted state-action value over the horizon from time t to $t + H$. This can be achieved by minimizing the following objective function:

$$J_\pi(\theta) = -\sum_{i=t}^{i=t+H} \lambda^{i-t} Q_\theta(s_i, \pi_\theta(s_i)) \tag{32}$$

3.4 Exploration strategy incorporating prior knowledge

In reinforcement learning, the exploration-exploitation trade-off is crucial. Exploitation refers to leveraging existing knowledge to take the seemingly optimal action to maximize immediate rewards, whereas exploration involves trying different actions to discover potentially better strategies that may yield higher long-term returns (Ladosz et al., 2022). Especially in the early stages of training, random exploration is commonly employed to collect interaction data. When the agent has little knowledge about the environment, random exploration enables it to try various actions and observe the outcomes, thereby gradually building an understanding of the environmental dynamics.

In MBRL, training requires sequential data. However, employing random exploration can lead to premature termination of training episodes, resulting in insufficient data for model training during the initial stages. Moreover, ineffective exploration in subsequent training phases can diminish learning efficiency and slow convergence. To address these challenges, an expert policy—provided by experienced practitioners or pretrained strategies—can be introduced during early training to guide the agent's behavior, facilitating effective data collection for model learning. Additionally, incorporating constraints based on prior knowledge during later training stages can help identify and rectify meaningless explorations, thereby enhancing learning efficiency and accelerating convergence.

To address the issue of insufficient training data in the early stages of MBRL, an expert policy derived from prior knowledge is employed to guide exploration. Specifically, during the first 30 training episodes, the agent interacts with the environment using an expert policy based on a pure tracking method, defined as Eqs. (33) and (34):

$$u_{a,i}^t = \alpha \Delta s_i^t + \beta \Delta q_i^t + 0.1 \times u_a^{\max^2} \tag{33}$$

$$u_{\delta,i}^t = \arctan\left(\frac{2L\sin\psi(t)}{l_d}\right) + 0.1 \times u_\delta^{\max^2} \tag{34}$$

where α and β are the weights for the distance error and velocity error, respectively. $l_d = \eta v + l_0$ denotes the look-ahead distance, η is the look-ahead time, l_0 represents the initial look-ahead distance, and ψ is the angle between the preview point and the vehicle. Noise is added to the actions to encourage exploration and improve policy generalization.

Crucially, during these episodes, the expert policy is used only for data collection to populate the replay buffer with high-quality transitions. Meanwhile, the RL agent is trained off-policy on this gathered data. By episode 31, the agent has therefore been effectively "warm-started", having learned a reasonable policy from expert demonstrations. This enables a seamless transition to agent-driven interaction without a significant drop in performance, despite the hard switch in data collection policy.

In addition, leveraging prior knowledge to impose action constraints throughout the training process helps prevent meaningless or unsafe exploration. In the context of lateral and longitudinal platoon control, specific corrective actions can be defined based on deviations in lateral distance and heading angle. For instance, if the lateral deviation exceeds 1 m and the heading angle deviation is greater than 0.25π radians, the vehicle should execute a right steer, corresponding to a negative steering angle. Conversely, if the lateral deviation is less than −1 m and the heading angle deviation is less than −0.25π radians, the vehicle should steer to the left, implying a positive steering angle. The mathematical expression of this action constraint is as Eq. (35):

$$u_\delta^i = \begin{cases} -|u_\delta^i|, & e^i > 1 \wedge \varphi^i > 0.25\pi \\ |u_\delta^i|, & e^i < -1 \wedge \varphi^i < -0.25\mathrm{i} \\ u_\delta^i, & \text{else} \end{cases} \tag{35}$$

The learning and planning parameters are summarized in Table 1. The learning parameters include the maximum steps per episode *Envs* and the number of episodes *Epis*. The planning parameters consist of the discount factor for evaluation γ, the discount factor of loss function λ, the number of planning iterations m, the planning horizon H, the number of sampled action sequences from distribution N, the number of sampled action sequences from policy N_π, the number of elite action sequences N', and the action distribution soft update coefficient σ.

Planning horizon selection (H = 3). In MPPI, we optimize an open-loop action sequence of length H in a receding-horizon manner. Increasing H improves look-ahead but also increases computation (more model rollouts) and amplifies multistep model errors. We set H = 3 as a compromise between real-time feasibility and prediction reliability. Importantly, the terminal value estimated by the critic network provides an approximation of the return beyond the H-step rollout, so the planning objective still accounts for long-term effects. Long-term string stability is therefore achieved by closed-loop replanning under the multi objective reward and is quantitatively evaluated by the cumulative suppression ratio i in Section 4.

Table 1 Learning and planning parameters

Symbol	Value	Symbol	Value
Envs	500	H	3
Epis	300	N	128
γ	0.99	N_π	25
λ	0.5	N'	32
m	3	σ	0.1
α	2	β	1.3

4 Experiments and results

This section is divided into three subsections. First, it presents the evaluation metrics and other comparative methods. Next, it analyzes the training process, focusing on the impact of incorporating prior knowledge into the MBRL training. Finally, it evaluates the control performance of different methods under speed-limited and real-data scenarios.

4.1 Evaluation metrics and comparative methods

The evaluation of vehicle platooning encompasses three primary metrics: the cumulative suppression rate ratio $d_{p,i}$, which assesses platoon stability; the inverse time to collision TTC_i^{-1}, which evaluates safety; and the operation time T, which measures computational efficiency.

The cumulative suppression rate ratio $d_{p,i}$ (Shi et al., 2022) evaluates the string stability of a vehicle platoon by measuring its ability to suppress traffic oscillations, as shown below:

$$d_{p,i} = \frac{\left(\sum_{t=0}^{s} |a_i^t - a_{i,\mathrm{mean}}|^2\right)^{\frac{1}{2}}}{\left(\sum_{t=0}^{s} |a_0^t - a_{0,\mathrm{mean}}|^2\right)^{\frac{1}{2}}} \tag{36}$$

where i denotes the index of the vehicle in the platoon, and s represents the total number of steps during platoon movement. a_0^t denotes the acceleration of the lead vehicle, a_i^t denotes the acceleration of vehicle i, and $a_{i,\mathrm{mean}}^t$ is the average acceleration of vehicle i over the s steps. When $d_{p,I} < 1$ and decreases as i increases, traffic oscillations weaken as they propagate upstream along the platoon, which means that the platoon is string stable.

The inverse of time to collision (TTC) (Balas and Balas, 2006) is used as a safety evaluation metric. Compared to TTC, it can more accurately quantify collision risks between vehicles and avoids the issue of infinite values when the speeds of the front and following vehicles are equal. The safety of vehicle i at time t is represented by $TTC_{i,t}^{-1}$, and its calculation method is as Eq. (37):

$$TTC_{i,t}^{-1} = \max\left(0, \frac{v_{i,t} - v_{i-1,t}}{p_{i-1,t} - p_{i,t}}\right) \tag{37}$$

When the value of $TTC_{i,t}^{-1}$ is 0, the vehicle has no collision risk and the highest level of safety. Moreover, the smaller the value is, the higher the safety.

By summing the values of $TTC_{i,t}^{-1}$ over the driving period of vehicle i, the cumulative TTC_t^{-1} can be obtained, which represents the overall safety during its entire driving time. It is calculated as Eq. (38):

$$TTC_i^{-1} = \sum_{t=0}^{n} \max\left(0, \frac{v_{i,t} - v_{i-1,t}}{p_{i-1,t} - p_{i,t}}\right) \tag{38}$$

The runtime T is an important metric for evaluating the response speed and computational complexity of different control methods in platooning tasks. By recording the time required for the platoon to complete the test under identical testing conditions and the same duration of the lead vehicle's movement, a shorter time indicates a faster response and higher control efficiency.

For comparison, the longitudinal-lateral decoupled platooning control method based on MPC in (Zhang et al., 2022) is adopted, along with the model-free reinforcement learning method using the twin delayed deep deterministic policy gradient (TD3) algorithm (Guo et al., 2017) for platooning control. Both methods utilize the same state space, action space, and reward function as those proposed in this study.

The parameters of the vehicles in the platooning environment include the number of vehicles in the platoon n, the desired

standstill spacing l, the vehicle length c, the vehicle wheelbase L, the desired time gap ε, the time step size dt, the maximum longitudinal acceleration control input $u_a^{\max}$, the maximum steering angle control input $u_\delta^{\max}$, the longitudinal control delay constant τ_a, and the lateral control delay constant τ_δ, as summarized in Table 2. These parameter values are determined based on previous studies and further fine-tuned through empirical testing.

4.2 Training results

In this study, the MBRL-based longitudinal-lateral control method for vehicle platooning, incorporating prior knowledge, is trained, and the variations in the average reward during the training process are analyzed. At the same time, this method is compared with the MBRL method using random exploration and the TD3 method based on MFRL through experimental comparisons. Specifically, both the MBRL method incorporating prior knowledge and the MBRL method using random exploration are trained for 300 episodes (500 steps per episode), while the TD3 method is trained for 500 episodes.

Fig. 4 illustrates the training progress and the variation in average reward for the three methods. The MBRL method incorporating prior knowledge converges after approximately 50,000 steps, whereas the MBRL method without prior knowledge still exhibits significant oscillations at 120,000 steps. In comparison, the TD3 method achieves stable convergence only after 130,000 steps. These results indicate that the learning efficiency of MBRL is superior to that of MFRL, and the integration of prior knowledge further enhances the convergence speed, enabling the agent to reach the optimal policy more rapidly. Moreover, in terms of reward stability, the MBRL method with prior knowledge demonstrates significantly smaller oscillation amplitudes than the other two approaches, reflecting a more stable training process. Compared to the MBRL baseline, which relies on random exploration, the prior knowledge-guided approach benefits from structured exploration and action constraints, thereby reducing ineffective exploration and training interruptions.

It should be noted that although the average reward curve shows visible fluctuations in the later training phase, this does not indicate policy instability. These fluctuations are primarily attributable to the stochastic nature of the exploration process and the sampling-based planning technique employed in the proposed framework. Specifically, a decaying noise is maintained throughout training to prevent premature convergence to local optima. Additionally, the inherent randomness in generating candidate action sequences during each planning iteration introduces minor variances in the cumulative reward. Despite these oscillations, the converged baseline confirms that the agent has successfully learned a robust control policy, which is further validated by its tracking performance in both speed-limit and real-world driving scenarios.

Table 2 Platoon environment parameters

Symbol	Value	Symbol	Value
n	5	dt	0.1 s
l	1 m	$u_a^{\max}$	3 m/s^2
c	5 m	$u_\delta^{\max}$	0.6 rad
L	3 m	τ_a	0.3 s
ε	1 s	τ_δ	0.3 s

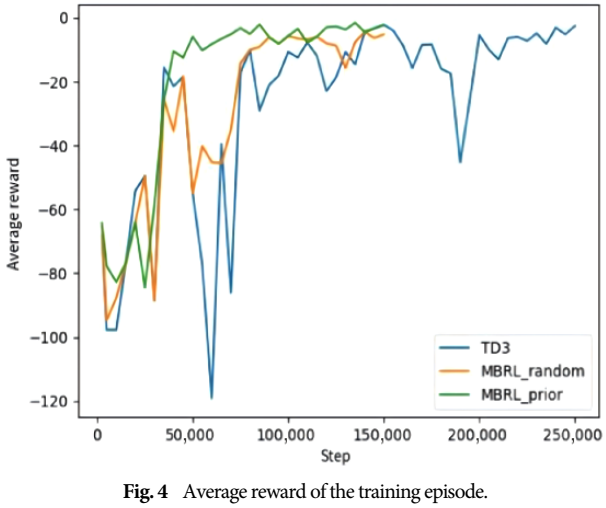


Fig. 4 Average reward of the training episode.

4.3 Test results

To comprehensively evaluate the effectiveness of the model-based reinforcement learning method with planning capability and prior knowledge for longitudinal-lateral control of vehicle platoons, this study conducts tests in both speed-limited and real-data scenarios. In the speed-limited scenario, the lead vehicle accelerates and decelerates according to speed limits, while in the real-data scenario, the longitudinal control of the lead vehicle follows real vehicle acceleration data from the NGSIM dataset (Montanino and Punzo, 2015).

These experiments enable a thorough evaluation of the performance of the MBRL method under different driving conditions. Comparisons are made with a decoupled MPC method and a TD3 longitudinal-lateral method, assessing their performance in terms of stability, safety, and computational efficiency. Due to the high inefficiency and instability caused by frequent ineffective and interrupted exploration during training, purely random exploration methods are not tested. The focus is placed on the effectiveness of the MBRL method integrated with prior knowledge.

In the speed-limited scenario, the lead vehicle follows a speed-limited control strategy to evaluate the control effects and platoon stability of different control methods. The trajectory of the lead vehicle and the driving states of the vehicle platoon in the curve are shown in Fig. 5.

The lateral distance deviation and heading angle deviation of vehicles under different methods in the speed-limited scenario are shown in Fig. 6. The MBRL method, by incorporating the environmental model for decision-making and planning, allows for more accurate adjustments to the lateral control strategy, thereby reducing lateral errors. In contrast, the MPC method

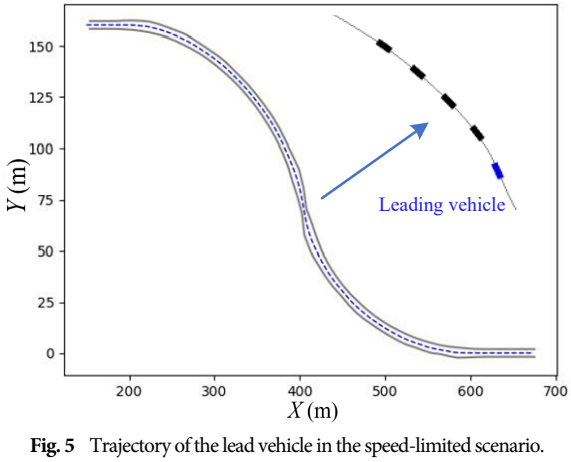


Fig. 5 Trajectory of the lead vehicle in the speed-limited scenario.

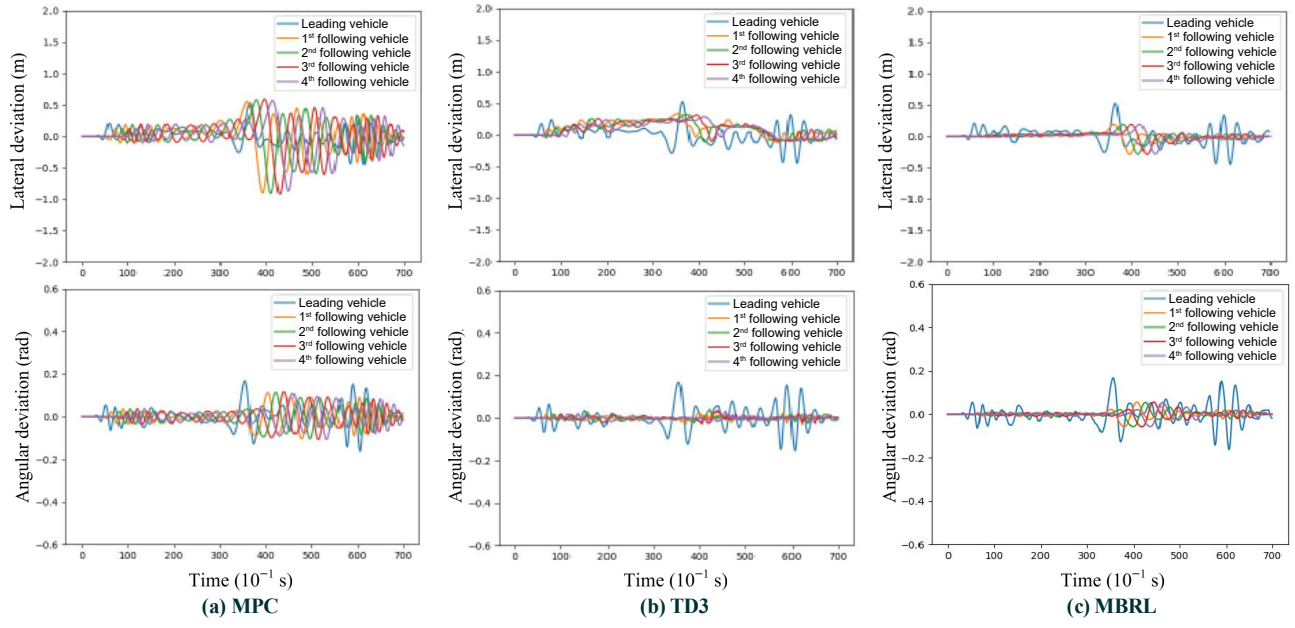


Fig. 6 Lateral distance deviation and heading angle deviation in the speed-limited scenario.

decouples the longitudinal and lateral control and approximates nonlinear state transitions to linear, which results in relatively large lateral deviations and heading angle deviations, with noticeable oscillations, when lateral disturbances occur in the lead vehicle. Although the TD3-based method can train an effective control strategy, its reliance solely on experience data for policy updates leads to a significant increase in lateral deviation, especially during sharp turns. In contrast, the MBRL method, by using the learned environmental model for planning, enables the agent to better adapt to complex lateral control tasks, effectively reducing lateral deviation and heading angle deviation and ensuring the platoon's tracking performance.

The position, speed, and acceleration of vehicles under different methods in the speed-limited scenario are shown in Fig. 7. The

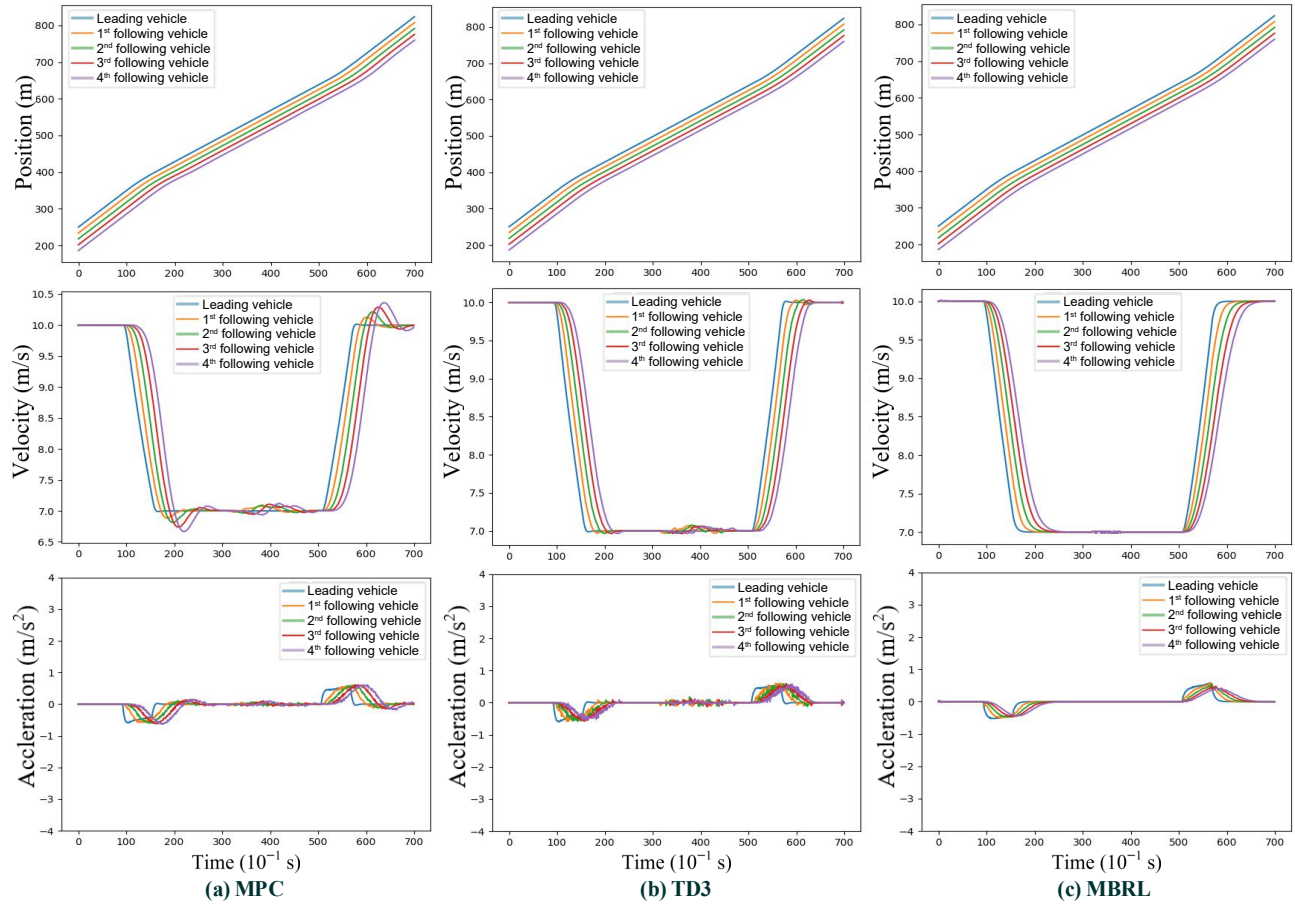


Fig. 7 Position, speed, and acceleration of the vehicle platoon in the speed-limited scenario.

position is given in the Frenet coordinate system, representing the length of the vehicle's projection point on the reference road. During the speed variation process, all three methods effectively control the platoon vehicles' speed. However, the MPC method exhibits relatively large fluctuations during the speed decrease phase. The TD3 method adjusts speed more smoothly and follows the lead vehicle's speed changes well, but from steps 350 to 450, due to significant changes in the road, the speed experiences some oscillations. The MBRL method's speed curve is the smoothest, indicating that it adapts to speed changes quickly and maintains stable control. In acceleration changes, both the MPC and TD3 methods show significant acceleration fluctuations during speed transitions, which affect comfort. The MBRL method's acceleration curve is the smoothest, showing that it adjusts speed more gently, providing a more comfortable driving experience. Overall, the MBRL method, which integrates the environmental model for planning, effectively enhances the control performance and comfort of the vehicle platoon.

The cumulative suppression rate ratio of the vehicle platoon under different methods in the speed-limited scenario is shown in Fig. 8. The cumulative suppression rate ratio for all methods is less than 1, indicating that the entire platoon exhibits string stability, meaning that traffic oscillations gradually diminish as they propagate through the platoon. Additionally, the MBRL method has a smaller value, indicating that it can more effectively suppress oscillations and improve the platoon's stability.

In the real data scenario, the longitudinal motion of the lead vehicle is based on the NGSIM data to evaluate the performance and adaptability in a real traffic environment. The trajectory of the lead vehicle and the driving states of the vehicle platoon in the curve are shown in Fig. 9.

The lateral distance deviation and lateral angle deviation of the vehicles under different methods in the real data scenario are shown in Fig. 10. In this scenario, the road curvature changes continuously, posing higher demands on the adaptability of lateral control strategies. For the MPC and TD3 methods, the lateral deviation accumulates when the road curvature changes before convergence, making it difficult to quickly recover to the ideal trajectory. In contrast, the MBRL method, by combining the environmental model for forward-looking planning, can adjust its control strategy in advance when the road curvature changes, preventing the continuous accumulation of lateral errors and improving lateral tracking ability and stability.

The cumulative suppression ratio of vehicle platoons under different methods in the real data-driven scenario is shown in Fig. 11. Compared to other methods, the proposed method

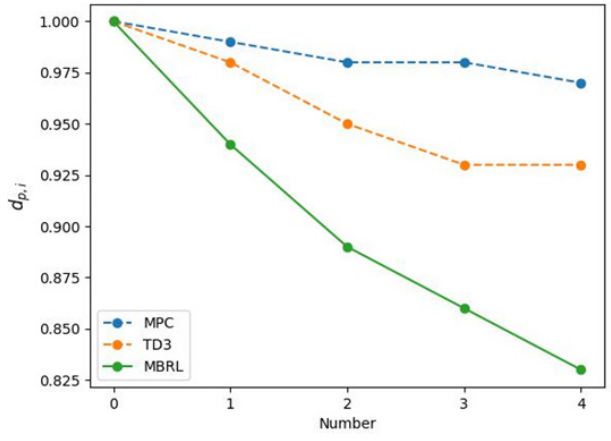


Fig. 8 Cumulative suppression ratio in the speed-limited.

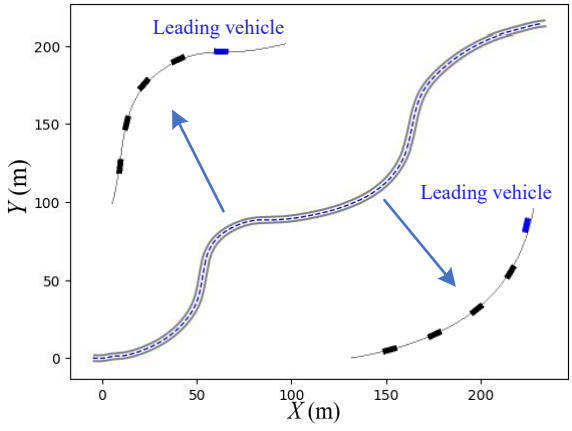


Fig. 9 Trajectory of the lead vehicle in the real-data scenario.

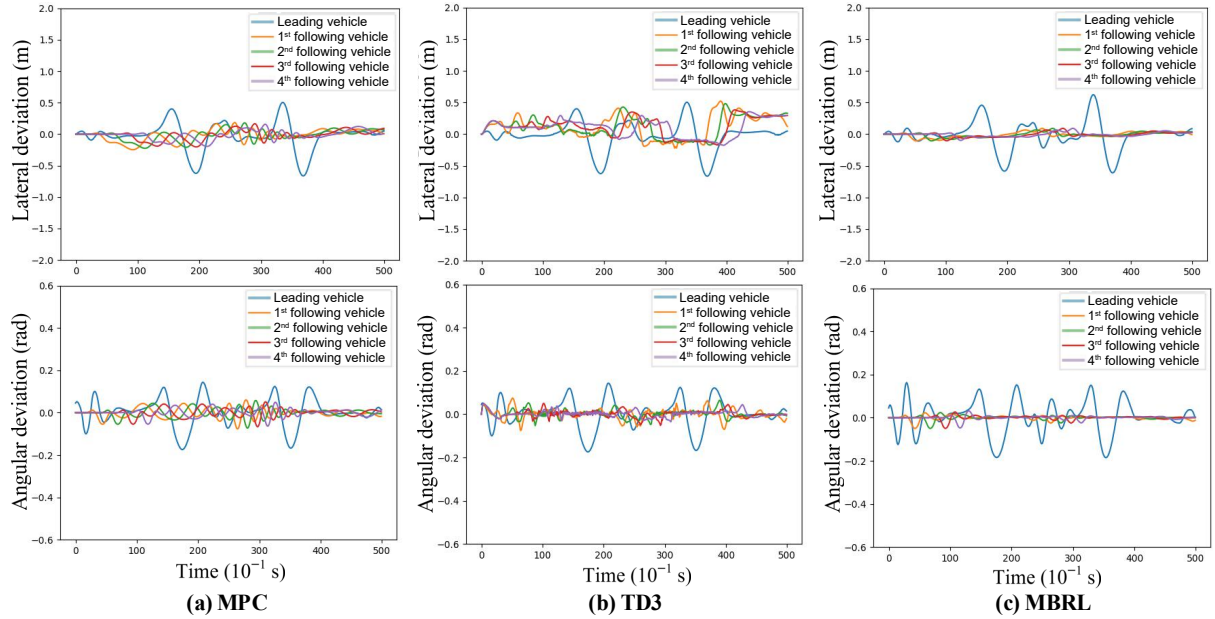


Fig. 10 Lateral distance deviation and heading angle deviation in the real-data scenario.

demonstrates a lower cumulative suppression ratio in this scenario, indicating its higher efficiency in suppressing platoon oscillations, thereby enhancing system stability and environmental adaptability.

The inverse time to collision and running time of different methods in the real lead vehicle data scenario are shown in Table 3. In this scenario, the TD3 method has a higher cumulative suppression ratio than the MPC method, mainly because the lead vehicle's acceleration changes are more abrupt. The TD3 method struggles to adapt to such abrupt acceleration variations, resulting in larger acceleration fluctuations for the following vehicles, which in turn increases the cumulative suppression ratio, negatively impacting the overall platoon stability. The MBRL method shows lower collision time reciprocal values across all vehicles, indicating higher safety for the vehicle platoon. The running time of the MBRL method (33.1 s) is between that of the MPC and TD3 methods, ensuring good control performance while maintaining favorable real-time capability. Overall, the proposed method performs better in convergence speed, longitudinal and lateral control effectiveness, platoon stability, safety, and applicability while ensuring computational efficiency.

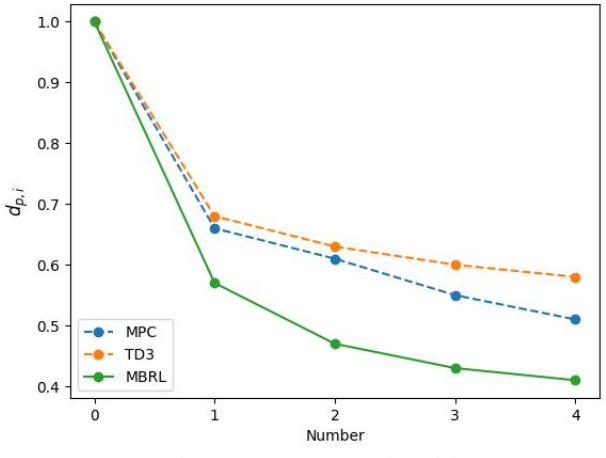


Fig. 11 Cumulative suppression ratio in the real-data scenario.

Table 3 Inverse time to collision and running time in the real-data scenario

Method	TTC_i^{-1}				T (s)
	1	2	3	4	
MPC	6.06	4.62	4.24	3.95	42.90
TD3	5.93	4.53	4.10	3.51	22.80
MBRL	4.15	3.19	2.91	2.74	33.10

5 Conclusions

This study proposes a prior-knowledge-integrated model-based reinforcement learning method with planning capabilities for vehicle platoon longitudinal-lateral control. The method achieves this by fusing policy and environment models for planning-driven decision-making and executing the first action of the optimized sequence in each cycle. It also designs a Frenet coordinate-based unified state space, a joint action space, and an integrated reward function that incorporates control performance, comfort, and safety. Additionally, the method introduces a prior-knowledge-guided exploration strategy to enhance learning efficiency and convergence. Simulation experiments in speed-limited and real lead-vehicle data scenarios validate its effectiveness, showing improved platoon lateral tracking, stability, safety, and adaptability while maintaining high computational efficiency. However, this study adopts simplifications such as a simplified vehicle kinematic model and perfect sensors and control systems. Future work addresses these issues by adopting more realistic complex vehicle models, considering sensor noise and communication delays, extending to multilane scenarios with advanced dynamic lane-change handling strategies, and enabling platoon control based on real-time dynamic vehicle information in dynamic communication environments.

Author contributions

Xia Wu: Writing – review & editing, Supervision, Resources, Funding acquisition, Methodology. **Haigen Min:** Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization, Supervision, Funding acquisition. **Zihao Mao:** Writing – original draft, review & editing, Validation, Software, Methodology, Formal analysis. **Yanbing Yan:** Writing – original draft, Validation, Methodology, Formal analysis. **Yang Liu:** Software, Data curation, Supervision. **Guoyuan Wu:** Formal Analysis, Data curation Project, administration.

Replication and data sharing

The data supporting the findings of this study are available at https://doi.org/10.26599/ETSD.2026.9190003.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Nos. U25B20136, 52372426, and 52302399); Fundamental Research Funds for the Central Universities, CHD (Nos. 300102243202 and 300102244713).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

Arulkumaran, K., Deisenroth, M. P., Brundage, M., Bharath, A. A., 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Process Mag*, **34**, 26–38.

Balas, V. E., Balas, M. M., 2006. Driver assisting by inverse time to collision. In: 2006 World Automation Congress, 1–6.

Barreno, F., Santos, M., Romana, M., 2025. A novel approach for self-driving vehicle longitudinal and lateral path-following control using the road geometry perception. *Electronics*, **14**, 1527.

Chatterjee, P., Khardon, R., 2025. Improving Planning and MBRL with Temporally Extended Actions. https://doi.org/10.48550/arXiv.2505.15754

Dong, K., Luo, Y., Wang, Y., Liu, Y., Qu, C., Zhang, Q., et al., 2024. Dyna-style model-based reinforcement learning with model-free policy optimization. *Knowl Based Syst*, **287**, 111428.

Elliott, D., Keen, W., Miao, L., 2019. Recent advances in connected and automated vehicles. *J Traffic Transp Eng Engl Ed*, **6**, 109–131.

Fujimoto, S., Hoof, H., Meger, D., 2018. Addressing Function Approximation Error in Actor-Critic Methods. In: Proceedings of the 35th International Conference on Machine Learning, 1587–1596.

Guanetti, J., Kim, Y., Borrelli, F., 2018. Control of connected and automated vehicles: State of the art and future challenges. *Annu Rev Control*, **45**, 18–40.

Guo, J., Luo, Y., Li, K., 2017. Adaptive fuzzy sliding mode control for coordinated longitudinal and lateral motions of multiple autonomous vehicles in a platoon. *Sci China Technol Sci*, **60**, 576–586.

Jiang, L., Xie, Y., Evans, N. G., Wen, X., Li, T., Chen, D., 2022. Reinforcement Learning based cooperative longitudinal control for reducing traffic oscillations and improving platoon stability. *Transp Res Part C Emerg Technol*, **141**, 103744.

Kotb, M., Weber, C., Wermter, S., 2023. Sample-efficient real-time planning with curiosity cross-entropy method and contrastive learning. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 9456–9463.

Ladosz, P., Weng, L., Kim, M., Oh, H., 2022. Exploration in deep reinforcement learning: A survey. *Inf Fusion*, **85**, 1–22.

Li, M., Cao, Z., Li, Z., 2021. A reinforcement learning-based vehicle platoon control strategy for reducing energy consumption in traffic oscillations. *IEEE Trans Neural Netw Learn Syst*, **32**, 5309–5322.

Li, X., Peng, F., Ouyang, Y., 2010. Measurement and estimation of traffic oscillation properties. *Transp Res Part B Methodol*, **44**, 1–14.

Li, Z., Chen, H., Liu, H., Wang, P., Gong, X., 2022. Integrated longitudinal and lateral vehicle stability control for extreme conditions with safety dynamic requirements analysis. *IEEE Trans Intell Transp Syst*, **23**, 19285–19298.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al., 2019. Continuous Control with Deep Reinforcement Learning. https://arxiv.org/abs/1509.02971

Liu, J., Shou, J., Fu, Z., Zhou, H., Xie, R., Zhang, J., et al., 2020. Efficient Reinforcement Learning Control for Continuum Robots Based on Inexplicit Prior Knowledge. https://arxiv.org/abs/2002.11573

Liu, X. Y., Wang, J. X., 2021. Physics-informed Dyna-style model-based deep reinforcement learning for dynamic control. *Proc R Soc A*, **477**, 20210618.

Moghadam, M., Alizadeh, A., Tekin, E., Elkaim, G. H., 2021. A deep reinforcement learning approach for long-term short-term planning on frenet frame. In: 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), 1751–1756.

Montanino, M., Punzo, V., 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transp Res Part B Methodol*, **80**, 82–106.

Pal, C. V., Leon, F., 2020. Brief survey of model-based reinforcement learning techniques. In: 2020 24th International Conference on System Theory, Control and Computing (ICSTCC), 92–97.

Pi, D., Xue, P., Xie, B., Wang, H., Tang, X., Hu, X., 2022. A platoon control method based on DMPC for connected energy-saving electric vehicles. *IEEE Trans Transp Electrif*, **8**, 3219–3235.

Qu, X., Yu, Y., Zhou, M., Lin, C. T., Wang, X., 2020. Jointly dampening traffic oscillations and improving energy consumption with electric, connected and automated vehicles: A reinforcement learning based approach. *Appl Energy*, **257**, 114030.

Ren, P., Jiang, H., Xu, X., 2023. Research on a cooperative adaptive cruise control (CACC) algorithm based on frenet frame with lateral and longitudinal directions. *Sensors*, **23**, 1888.

Shan, Z., Zhao, J., Zhu, B., Lv, C., Zhao, Y., Ge, L., et al., 2024. Safe and efficient trajectory planning considering longitudinal and lateral coupled limits. *IEEE Trans Veh Technol*, **73**, 10714–10719.

Shi, H., Zhou, Y., Wang, X., Fu, S., Gong, S., Ran, B., 2022. A deep reinforcement learning-based distributed connected automated vehicle control under communication failure. *Comput Aided Civ Infrastruct Eng*, **37**, 2033–2051.

Su, Y., Yang, X., Shi, P., Wen, G., Xu, Z., 2024. Consensus-based vehicle platoon control under periodic event-triggered strategy. *IEEE Trans Syst Man Cybern Syst*, **54**, 533–542.

Wang, C., Gong, S., Zhou, A., Li, T., Peeta, S., 2020. Cooperative adaptive cruise control for connected autonomous vehicles by factoring communication-related constraints. *Transp Res Part C Emerg Technol*, **113**, 124–145.

Wang, P., Deng, H., Zhang, J., Wang, L., Zhang, M., Li, Y., 2022. Model predictive control for connected vehicle platoon under switching communication topology. *IEEE Trans Intell Transp Syst*, **23**, 7817–7830.

Yan, M., Ma, W., Zuo, L., Yang, P., 2019. Dual-mode distributed model predictive control for platooning of connected vehicles with nonlinear dynamics. *Int J Control Autom Syst*, **17**, 3091–3101.

Yan, R., Jiang, R., Jia, B., Huang, J., Yang, D., 2022. Hybrid car-following strategy based on deep deterministic policy gradient and cooperative adaptive cruise control. *IEEE Trans Autom Sci Eng*, **19**, 2816–2824.

Yavas, U., Kumbasar, T., Ure, N. K., 2022. Model-based reinforcement learning for advanced adaptive cruise control: A hybrid car following policy. In: 2022 IEEE Intelligent Vehicles Symposium (IV), 1466–1472.

Yu, M., Evangelou, S. A., Dini, D., 2024. Advances in active suspension systems for road vehicles. *Engineering*, **33**, 160–177.

Zhang, Y., Wu, Z., Zhang, Y., Shang, Z., Wang, P., Zou, Q., et al., 2022. Human-lead-platooning cooperative adaptive cruise control. *IEEE Trans Intell Transp Syst*, **23**, 18253–18272.

Zhang, Z., Wang, C., Zhao, W., Feng, J., 2022. Longitudinal and lateral collision avoidance control strategy for intelligent vehicles. *Proc Inst Mech Eng Part D J Automob Eng*, **236**, 268–286.

Zhu, Y., Wu, J., Su, H., 2022. V_2V-based cooperative control of uncertain, disturbed and constrained nonlinear CAVs platoon. *IEEE Trans Intell Transp Syst*, **23**, 1796–1806.

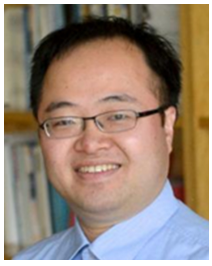
Xia Wu received the Ph.D. degree in traffic information engineering and control from Chang'an University, China, in 2020. Her research interests include mixed traffic flow control, trajectory optimization in a mixed traffic flow with connected and automated vehicles, and human-driven vehicles.

Haigen Min received the Ph.D. degree in traffic information engineering and control from Chang'an University, China, in 2018. He is currently a Professor at Chang'an University. His research interests include high-precision localization, environment perception, and cooperative control for connected and autonomous vehicles, and their fault detection and diagnosis.

Zihao Mao received the B.E degree in Internet of Things engineering from Yangtze University, China, in 2024. He is currently pursuing the M.S. degree at Chang'an University. His research interests include cooperative control of intelligent connected vehicle platoons

Yanbing Yan received the B.E. degree in Internet of Things engineering in 2022 from Chang'an University, China, where he is currently pursuing the M.S degree at Chang'an University. His research interests include cooperative control for intelligent connected vehicles and reinforcement learning.

Yang Liu received the B.S. degree in Computing (Hons.) from the National University of Singapore (NUS) in 2005, and the Ph.D. degree in 2010. He started his post doctoral work with NUS, MIT, and SUTD. In 2012 fall, he joined Nanyang Technological University (NTU) as a Nanyang Assistant Professor. He is currently a Full Professor and the Director of the Cybersecurity Lab, NTU. His research has bridged the gap between the theory and practical usage of formal methods and program analysis to evaluate the design and implementation of software for high assurance and security. His work led to the development of a state-of-the-art model checker, Process Analysis Toolkit (PAT). With more than 20 million Singapore dollar funding support, he is leading a large research team working on the state-of-the-art software engineering and cybersecurity problems. He has authored or coauthored more than 300 publications and six best paper awards in top-tier conferences and journals. His research interests include software verification, security, and software engineering. In 2011, he was the recipient of the Temasek Research Fellowship at NUS to be the Principal Investigator in the area of cyber security.



Guoyuan Wu (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2010. He holds a Full Researcher and Adjunct Professor position with the Bourns College of Engineering — Center for Environmental Research & Technology (CE—CERT) and Department of Electrical & Computer Engineering, University of California at Riverside, Riverside, CA, USA. His research interests include the development and evaluation of sustainable and intelligent transportation system (SITS) technologies, including connected and automated transportation systems (CATSs), shared mobility, transportation electrification, optimization and control of vehicles, traffic simulation, and emissions measurement and modeling. His is an Associate Editor for a few journals, including *IEEE Transactions on Intelligent Transportation Systems*, *SAE International Journal of Connected and Automated Vehicles*, and *IEEE Open Journal of ITS*. He is also a Member of a few Standing Committees of the Transportation Research Board. He was the recipient of 2020 Vincent Bendix Automotive Electronics Engineering Award and 2021 Arch T. Colwell Merit Award.