Commun Transp Res

Research Article

Physics-informed platooning with learning-augmented calibration and compensation: a real-world study

Chengqi Liu[1,†], Qiang Ma[2,†], Xiwu Wang[2], Qiang Sun[1], Yinke Sun[3], Zhiyuan Liu[1,4], Nan Zheng[5], Kai Huang[2,✉]

Cite this article: https://doi.org/10.26599/COMMTR.2026.9640049

ABSTRACT: Autonomous vehicle platooning improves overall energy efficiency and enhances road capacity through close-range cooperative driving. However, practical deployment under complex real-world conditions remains challenging due to sensor noise, model uncertainties, and actuator nonlinearities. This paper presents a hierarchical control framework that integrates multi-source sensor fusion and a controller combining longitudinal Model Predictive Control (MPC) with Lateral Feedforward-Feedback Control (LFFC) with data-driven improvement. First, a prediction-correction fusion module combines wheel odometry (ODOM), Inertial Measurement Unit (IMU), and GPS measurements to obtain continuous and globally consistent vehicle state estimates. Second, an MPC-LFFC-based physical controller is developed, where longitudinal MPC optimizes velocity corrections and lateral LFFC combines curvature feedforward with proportional feedback of heading and lateral errors. Third, a data-driven improvement module comprises an offline Conservative Q-Learning (CQL) agent that automatically calibrates controller parameters, and a neural residual learning network that predicts feedforward compensation. Extensive experiments on a real-world vehicle platooning platform demonstrate that the proposed framework consistently outperforms baselines, achieving a position RMSE of 0.0781 m and a 30.66% improvement in linear velocity tracking.

KEYWORDS: Residual Learning; Conservative Reinforcement Learning; Model Predictive Control; Autonomous Platooning; Robust Formation Control

1 Introduction

Vehicle platooning, in which multiple vehicles travel cooperatively at reduced inter-vehicle gaps, has been widely recognized as an effective means to improve fuel efficiency, reduce aerodynamic drag, and enhance traffic throughput, playing a pivotal role in the evolution of intelligent transportation systems (Liu et al., 2025; Wang et al., 2025b). Compared with isolated driving, platooning offers substantial benefits, particularly for heavy-duty vehicles, where energy savings from reduced air resistance are most pronounced (Zhang et al., 2020; Ding et al., 2022; Zhang et al., 2022). With the advent of vehicle-to-vehicle (V2V) communication, cooperative adaptive cruise control (CACC) further enables shorter safe headways while preserving string stability, thereby mitigating disturbance propagation and improving overall traffic efficiency (Xie et al., 2024; Lin et al., 2024).

Unlike single-vehicle trajectory tracking, platoon control requires each following vehicle to simultaneously satisfy safety, comfort, and real-time constraints while responding to the leader's motion and maintaining a desired spatial configuration (Rebelo et al., 2024; Tian et al., 2026a). This multi-layered requirement spanning perception, decision, and actuation introduces multiple sources of error that propagate through the system and degrade formation stability (Wang et al., 2025c). In practice, these errors can be broadly attributed to three interconnected aspects: localization uncertainty, tracking control inaccuracies, and hardware-induced actuation discrepancies.

Localization uncertainty arises from sensor limitations and environmental disturbances. GPS and Global Navigation Satellite Systems (GNSS) suffer from occlusion and multipath effects, leading to unstable absolute position estimates, while IMUs provide continuous motion information but accumulate drift over time. Cameras and Light Detection and Ranging (LiDAR) sensors offer complementary local accuracy, yet their performance remains highly dependent on lighting conditions, texture richness, and geometric structure (Joubert et al., 2020). Since no single sensor can satisfy the requirements of platoon control, multi-sensor fusion has become a key solution. Existing studies broadly fall into two categories: one focuses on unified perception representations such as bird's-eye view (BEV) modeling and multimodal fusion for autonomous driving (Ping et al., 2025; Jiang et al., 2026); the other concentrates on vehicle-state estimation and localization, aiming to mitigate GNSS blockage, IMU drift, and visual/LiDAR degradation through sensor fusion (Alaba, 2024; Cheng et al., 2024; Liu et al., 2023a), with extensions

† Chengqi Liu and Qiang Ma contributed equally to this work.

[1] Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, Nanjing 211189, China. [2] School of Instrument Science and Engineering, State Key Laboratory of Comprehensive PNT Network and Equipment Technology, Southeast University, Nanjing 210096, China. [3] College of Artificial Intelligence, Nankai University, Tianjin 300071, China. [4] School of Statistics and Data Science, Southeast University, Nanjing 211189, China. [5] Department of Civil Engineering, Monash University, Melbourne VIC 3800, Australia.

✉ Corresponding author. E-mail: kaihuang@seu.edu.cn

Received: July 5, 2026; Revised: August 18, 2026; Accepted: August 20, 2026

toward robustness under adverse environments (Vinoth and Sasikumar, 2024). However, in platooning scenarios, localization errors manifest not only as absolute pose drift but also as inconsistencies in relative position, heading, and formation geometry among vehicles. Such errors directly enter the control loop, leading to overcorrection, delayed responses, or formation distortion (Huang et al., 2022). Existing methods predominantly address single-vehicle pose estimation and perception consistency (Zhang et al., 2026), and do not sufficiently account for the formation-level propagation of localization errors.

Tracking control inaccuracies constitute another major challenge. The discrepancy between desired and actual inter-vehicle spacing is a central performance metric in platooning and is typically addressed through control design. Linear controllers such as Proportional-Integral-Derivative (PID) and LQR are widely used due to their simplicity and computational efficiency (Ma et al., 2020; Chu et al., 2022). MPC has gained significant attention for its receding-horizon optimization framework and explicit constraint handling (Zhang et al.,2024), with extensions including distributed MPC for heterogeneous platoons, economic optimization, coordination on curved roads s (Gratzer et al., 2024; Li et al., 2024a; Gao et al., 2025; Zuo et al., 2025). Learning-based control methods leverage data-driven models to adapt to complex and nonlinear environments s (Guo et al., 2022; Ma et al., 2024), yet their lack of interpretability remains a limitation. Hybrid approaches, attempt to combine physical models with neural networks, either embedding physical priors into learning models or combining model-based and learned outputs s (Li et al., 2022; Zhang and Du, 2023; Cheng et al., 2025; Tian et al., 2026b). However, many of these methods tend to approximate full system dynamics from scratch, requiring large-scale data and potentially underutilizing established physical models (Ma et al., 2025). Reinforcement learning (RL) has also been applied to optimize control strategies without explicit models (Gao et al., 2024; Wang et al., 2023a.; Lai et al., 2024), and multi-agent RL further captures coupled interactions among vehicles (Dai et al., 2024; Wu et al., 2026). Nevertheless, RL-based platoon control integrated with physical models remains relatively underexplored (Farimani et al., 2026).

Moreover, while traditional car-following models primarily consider longitudinal errors, real-world driving frequently involves curved roads where lateral deviation and heading misalignment become non-negligible (Liu et al., 2023b; Wang et al., 2026). Some studies have considered coupled lateral–longitudinal dynamics (Feng et al., 2023; Chen et al., 2025), but fully coupled controllers often entail high computational complexity, which may introduce latency and degrade closed-loop stability. Therefore, a lightweight yet effective decoupled control structure that still accounts for lateral–longitudinal coupling is highly desirable.

Hardware-induced errors further undermine the performance of model-based controllers in practice. While simulation environments often assume idealized actuation, real vehicle platooning systems are significantly affected by actuator dynamics, chassis mechanics, power systems, communication delays, and onboard computation constraints (Li et al., 2024b; Ma et al., 2025; Wang et al., 2025a). Even when control commands are optimal at the algorithmic level, hardware response delays and nonlinearities introduce lag and attenuation, leading to degraded platoon stability (Wang et al., 2023b; Zhang et al., 2025). Real-world actuators such as motors, steering systems, and brakes exhibit inherent response delays and physical constraints (Yang et al., 2023). In addition, battery voltage variation, road conditions, payload distribution, tire wear, and manufacturing inconsistencies can cause significant variability in actuation responses across vehicles and operating conditions. Residual learning approaches have demonstrated that combining model-based control with learning-based compensation can improve adaptability while preserving stability (Trumpp et al., 2023). Related studies further show that physics-enhanced predictive control and learning-based robust control can effectively mitigate actuator delay, model uncertainty, and stochastic latency effects (Shi and Zhang, 2022; Long et al., 2024). However, most existing compensation methods rely on idealized state feedback or simulation-generated data, and they do not sufficiently address the cumulative impact of hardware-induced errors in multi-vehicle platooning systems.

Motivated by these limitations, we propose a new platooning control architecture that integrates multi-source state fusion, a model-based MPC-LFFC controller, and a data-driven improvement module. First, sensor information is fused to align leader and follower states in a unified coordinate system, enabling consistent formation feedback computation. Second, the model-based controller adopts a decoupled lateral–longitudinal structure, where longitudinal MPC generates velocity corrections while lateral LFFC combines curvature feedforward with proportional error feedback to ensure trajectory tracking, achieving a balance between computational efficiency and real-time feasibility. Finally, the data-driven module employs CQL for offline calibration of key control parameters and a neural residual model to estimate the discrepancy between nominal control commands and actual chassis responses. This design preserves the interpretability and constraint-handling capability of the model-based controller while improving robustness against sensor noise, unmodeled dynamics, and actuator nonlinearities in real-world platooning systems.

The remainder of this paper is organized as follows. Section 2 formulates the platooning control problem and presents the overall system architecture. Sections 3, 4, and 5 detail the proposed multi-source state fusion, the MPC-LFFC control structure, and the data-driven calibration and compensation methods, respectively. Section 6 presents real-world vehicle experimental results. Finally, Section 7 concludes the paper and discusses future work.

2 Overall framework

The proposed control architecture adopts a physics-informed and data-driven paradigm. Without loss of generality, we consider a multi-vehicle platooning scenario. Each vehicle is equipped with multiple sensors, such as GPS, IMU, and wheel odometry, and each vehicle has access to the global route plan. In Fig. 1, the green vehicle denotes the leader, and the blue vehicle denotes the follower. The leader and followers can communicate with each other via V2V technology.

As illustrated in Fig. 1, at each control cycle, the follower first calibrates the positions of both the leader and itself through the Multi-Source State Fusion module, and then evaluates the current platoon states and relative errors with respect to the given trajectory. Subsequently, these errors are fed into the Physical Model Controller, which employs a decoupled longitudinal–lateral control structure to generate the nominal desired velocity $\mathbf{u}_{\text{nom}}$. The parameters of this model are adjusted in real time by the CQL-based calibration method. Meanwhile, the Residual Learning Model estimates the discrepancy between the actual chassis-executable velocity and the nominal command based on current operating conditions, and outputs a compensation term $\Delta\mathbf{u}_{\text{nn}}$. The final velocity command $\mathbf{u}_{\text{cmd}}$ executed by the chassis is the sum of the nominal and compensation terms.

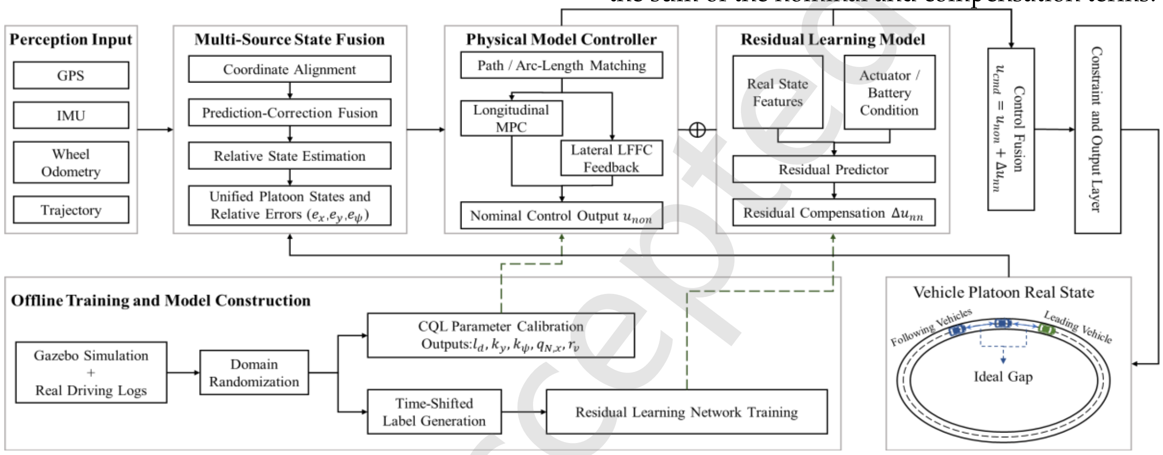

Figure 1: Overall framework of the proposed method.

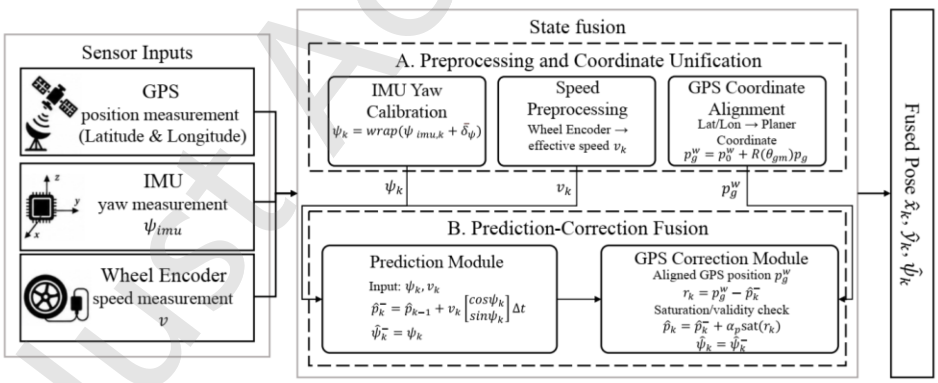

Figure 2: Multi-source data fusion framework for state estimation.

To enable real-time parameter assignment and compensation during deployment, the data-driven models must be pre-trained offline. As shown in Fig. 1, we construct datasets from both Gazebo simulations and real-world operation logs, which are then used separately for training the CQL-based calibration agent and the Residual Learning Model.

The following sections describe in detail the sensor data fusion method, the physical MPC-LFFC controller, and the data-driven calibration and residual learning modules, respectively.

3 Multi-source state fusion

This paper fuses information from GPS, IMU, and wheel odometry to construct the vehicle pose. The overall framework is illustrated in Fig. 2. Wheel odometry provides continuous longitudinal velocity in the body frame, while the IMU supplies heading measurements. The position obtained by dead-reckoning from these two sources accumulates drift over time. GPS provides global position references, but its observation noise and occasional outliers preclude direct substitution for the propagated state. Given these complementary characteristics, the vehicle state fusion adopts a prediction-correction structure: the vehicle position is continuously propagated using odometry velocity and calibrated IMU heading, and the propagated result is then corrected in a weighted and bounded manner using the coordinate-aligned GPS observations. This yields a continuous vehicle position and heading in a unified navigation frame with global position constraints, which are subsequently used for reference point construction, tracking error computation, and control law derivation.

3.1. GPS local coordinate transformation and alignment

Vehicle state modeling, tracking error characterization, and control law design are all based on a unified local coordinate description. To incorporate GPS measurements into the state estimation, the latitude and longitude readings are first converted to a local planar coordinate system and subsequently mapped to the navigation coordinate frame adopted by the controller. Let $\{g\}$ denote the GPS local planar frame, and $\{w\}$ denote the unified navigation frame (world frame). Let $\mathbf{p}_g = [x_g, y_g]^T$ denote the position in the GPS local frame, and $\mathbf{p}_g^w = [x_g^w, y_g^w]^T$ denote the corresponding position in the unified navigation frame. Since there generally exist a translational offset and a heading angle between the two coordinate frames, the alignment is accomplished using a planar rotation and translation:

$$\begin{bmatrix} x_g^w \\ y_g^w \end{bmatrix} = \begin{bmatrix} x_0^w \\ y_0^w \end{bmatrix} + \mathbf{R}(\theta_{gw}) \begin{bmatrix} x_g \\ y_g \end{bmatrix} \tag{1}$$

where (x_0^w, y_0^w) denotes the position of the GPS local origin in the unified navigation frame, and θ_{gw} denotes the rotation angle from the GPS local frame to the unified navigation frame. This rotation angle can be obtained through initial calibration motions, extrinsic parameter calibration, or coordinate alignment procedures of multi-sensor localization systems. $\mathbf{R}(\theta_{gw})$ is the two-dimensional rotation matrix defined as

$$\mathbf{R}(\theta_{gw}) = \begin{bmatrix} \cos\theta_{gw} & -\sin\theta_{gw} \\ \sin\theta_{gw} & \cos\theta_{gw} \end{bmatrix} \tag{2}$$

Through the above transformation, the local position observations from GPS are mapped to the unified navigation frame, thereby maintaining consistency with the odometry, IMU, and vehicle state variables used in subsequent path tracking control.

3.2. IMU heading alignment

The IMU provides continuous heading information for vehicle state fusion. However, due to mounting orientation, initial zero offsets, and coordinate frame definitions, there typically exists a constant bias between its output and the unified navigation frame. Therefore, during the initialization phase, the IMU output is calibrated using the reference heading in the unified navigation frame. The reference heading is used to establish the heading coordinate baseline, but its continuous output may be affected by local positioning links, filtering processes, and update delays; thus, it is not directly used as the heading input during operation. After the bias calibration is completed, the IMU continuously supplies the vehicle heading. Let the reference heading for the i -th calibration sample be denoted as $\psi_{r,i}$, and the IMU heading output as $\psi_{\mathrm{imu},i}$. Considering the periodicity of angular variables and measurement noise, the heading calibration offset is determined via the circular mean:

$$\bar{\delta}_\psi = \operatorname{atan2}\left(\sum_{i=1}^{N_\psi} \sin\left(\operatorname{wrap}(\psi_{r,i} - \psi_{\mathrm{imu},i})\right), \sum_{i=1}^{N_\psi} \cos\left(\operatorname{wrap}(\psi_{r,i} - \psi_{\mathrm{imu},i})\right) \right) \tag{3}$$

Here, N_ψ denotes the number of heading calibration samples, $\bar{\delta}_\psi$ is the calibration offset of the IMU heading relative to the unified navigation frame, and $\operatorname{wrap}(\cdot)$ normalizes the angle to the range $[-\pi, \pi]$. The reference heading is only used to determine the coordinate offset; it does not represent the desired heading for the controller, nor does it serve as the ground-truth vehicle heading. After calibration, the vehicle heading at the k -th fusion cycle is given by $\psi_k = \operatorname{wrap}(\psi_{\mathrm{imu},k} + \bar{\delta}_\psi)$.

3.3. Odometry prediction and GPS-constrained correction

Based on the above multi-sensor observations, the vehicle state fusion adopts a prediction-correction framework, which consists of the following two stages:

(1) Prediction: At the k -th fusion cycle, the vehicle position is denoted as $\hat{\mathbf{p}}_k = [\hat{x}_k, \hat{y}_k]^T$, and the heading as $\hat{\psi}_k$. Using the fused position from the previous cycle, the odometry linear velocity v_k , and the calibrated heading ψ_k , the position prediction for the current cycle is given by

$$\hat{\mathbf{p}}_k^- = \hat{\mathbf{p}}_{k-1} + v_k \begin{bmatrix} \cos\psi_k \\ \sin\psi_k \end{bmatrix} \Delta t_k, \qquad \hat{\psi}_k^- = \psi_k \tag{4}$$

where Δt_k is the fusion period, $\hat{\mathbf{p}}_k^-$ and $\hat{\psi}_k^-$ are the position and heading predictions before incorporating the current GPS observation. To suppress position drift caused by odometry noise at low speeds, the prediction update is scaled by a velocity-dependent confidence factor $\beta_k = \min(|v_k| / v_{\min}, 1)$ instead of being completely disabled; i.e., the incremental displacement is multiplied by β_k . This prevents the state from being frozen and maintains smooth transitions.

(2) Correction: The coordinate-transformed GPS position observation is denoted as $\mathbf{p}_{g,k}^w = [x_{g,k}^w, y_{g,k}^w]^T$. In the correction stage, the correction magnitude c_k is determined based on the discrepancy between the GPS observation and the predicted position. Observations with overly short update intervals or discrepancies exceeding the rejection threshold are discarded; for moderate discrepancies, the correction magnitude is appropriately reduced; remaining valid observations are used for state update within a given upper bound. The resulting fused position is given by

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_k^- + \alpha_p \operatorname{sat}_{c_k}\left(\mathbf{p}_{g,k}^w - \hat{\mathbf{p}}_k^-\right), \qquad \hat{\psi}_k = \hat{\psi}_k^- \tag{5}$$

where α_p is the GPS correction weight, and $\operatorname{sat}_{c_k}(\cdot)$ denotes a vector saturation operator with bounded norm. GPS participates in the position update only as a bounded correction term, rather than directly replacing the predicted state, thereby mitigating the impact of abnormal observations on the continuity of the fused state.

4 MPC-LFFC based physical model

During platoon tracking, longitudinal and lateral motions exhibit different control objectives and dynamic characteristics. To avoid the complexity of a full two-dimensional online optimization, this study adopts a decoupled longitudinal-lateral control structure. At each sampling step, the controller computes the longitudinal error e_x , lateral error e_y , and heading error e_ψ from the desired tracking state and the current vehicle pose. The longitudinal channel optimizes the linear velocity command using finite-horizon MPC, while the lateral channel generates the angular velocity command through curvature feedforward proportional feedback of lateral and heading errors. This structure reduces the optimization complexity and allows the linear and angular velocity responses to be tuned separately.

4.1. Longitudinal MPC control

The longitudinal controller regulates the tracking error along the desired moving direction by optimizing the velocity correction over a finite prediction horizon. At the k -th control step, the velocity correction sequence is denoted as $\Delta V_k = \left[\Delta v_{0|k}, \Delta v_{1|k}, \ldots, \Delta v_{N-1|k}\right]^T$, where N is the prediction horizon. Given the baseline velocity v_b , the predicted velocity and the longitudinal error evolution are formulated as

$$\begin{aligned} v_{j|k} &= v_b + \Delta v_{j|k} \\ e_{x,j+1|k} &= e_{x,j|k} - T_s \Delta v_{j|k} \end{aligned} \tag{6}$$

where $j = 0,1,\ldots,N-1$, T_s is the control sampling period, and $e_{x,0|k} = e_x(k)$ is initialized by the current longitudinal error. This prediction model describes how the velocity correction affects the future longitudinal tracking error.

Based on the above prediction model, the longitudinal MPC controller determines the optimal velocity correction sequence by solving the following finite-horizon optimization problem:

$$\Delta V_k^* = \arg\min_{\Delta V_k} J(k,N) \tag{7}$$

The finite-horizon objective penalizes both the predicted longitudinal error and the magnitude of velocity correction. The cost function is defined as:

$$J(k,N) = \sum_{j=0}^{N-1} q_x e_{x,j|k}^2 + q_{N,x} e_{x,N|k}^2 + \sum_{j=0}^{N-1} r_v \Delta v_{j|k}^2 \tag{8}$$

where the first summation includes the initial error $e_{x,0|k}$ (which is fixed but still penalized for consistency), and the terminal error is separately weighted. The optimization is subject to the velocity and correction bounds:

$$\begin{aligned} & v_{\min} \le v_b + \Delta v_{j|k} \le v_{\max} \\ & \Delta v_{\min} \le \Delta v_{j|k} \le \Delta v_{\max} \\ & j = 0,1,\ldots,N-1 \end{aligned} \tag{9}$$

Since the prediction model is linear, the objective is quadratic, and the constraints are linear bounds, the resulting optimization problem is a convex quadratic programming (QP) problem that can be solved efficiently.

After solving the finite-horizon problem, the first optimized velocity correction is used to construct the longitudinal command:

$$v_{\text{nom}}(k) = \text{clip}\left(v_b + \Delta v_{0|k}^*, v_{\min}, v_{\max}\right) \tag{10}$$

At the next sampling step, the controller recalculates the longitudinal error using the updated vehicle state and solves the QP problem again. Compared with a one-shot planning strategy that determines the entire control sequence in advance, the receding-horizon optimization scheme continuously incorporates new state observations and updates subsequent control inputs according to the actual vehicle response, thereby improving the adaptability of the longitudinal controller to disturbances, modeling errors, and state-estimation uncertainties.

4.2. Lateral Feedforward-Feedback Control (LFFC)

The lateral controller generates the angular velocity command according to the lateral position error and heading error between the current vehicle state and the desired tracking state. A feedforward--feedback structure is adopted, where the curvature feedforward term represents the nominal angular velocity required by the desired motion curvature, and the feedback term compensates for real-time tracking errors using a proportional structure. Let e_y and e_ψ denote the lateral position error and heading error, respectively. The angular velocity command is given by

$$\omega_{\text{nom}} = \text{sat}_\omega\left(v_{\text{nom}}\kappa_d + k_y e_y + k_\psi e_\psi\right) \tag{11}$$

where $v_{\text{nom},k}$ and $\omega_{\text{nom},k}$ denote the linear and angular velocity commands output by the controller, κ_d is the desired motion curvature, and k_y and k_ψ are the feedback gains corresponding to the lateral position error and heading error, respectively. The operator $\text{sat}_\omega(\cdot)$ denotes the angular velocity saturation function.

5 Data-driven improvement

Although the MPC-LFFC controller provides a principled model-based framework for vehicle platooning control, its practical performance is constrained by two sources of uncertainty. First, its effectiveness relies heavily on manually selected parameters; second, the underlying kinematic model inevitably neglects unmodeled nonlinearities such as actuator delays, tire slip, motor dead zones, and battery-voltage-dependent dynamics, causing persistent deviations between the commanded and actual vehicle states (Huo et al., 2025). To address these limitations, we introduce a data-driven improvement framework that operates at two complementary levels in decision and control procedures, as shown in Fig. 3.

5.1. Reinforcement learning for parameter calibration

The first level addresses the inherent difficulty of optimal parameter selection. An offline reinforcement learning agent is designed to automatically calibrate the core controller parameters from historical driving data, adapting them to different operating conditions while preserving the interpretable structure of the baseline controller.

To optimize these parameters from historical driving logs, the calibration process is formulated as a Markov Decision Process (MDP), defined by the tuple $\mathrm{M} = \langle \mathrm{S}, \mathrm{A}, \mathrm{P}, \mathrm{R}, \gamma \rangle$, where the autonomous vehicle acts as the agent interacting with the V2V tracking environment. Here, S denotes the state space, A the action space, $\mathrm{P}: \mathrm{S} \times \mathrm{A} \to \Delta(\mathrm{S})$ the state transition probability (where $\Delta(\mathrm{S})$ is the set of probability distributions over S), i.e., $\mathrm{P}(s'|s,a)$ gives the probability of transitioning to s' given s and a , $\mathrm{R}: \mathrm{S} \times \mathrm{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0,1)$ the discount factor.

At each decision step t , the agent observes a state vector $\mathbf{s}(t) \in \mathrm{S}$ that captures the current tracking condition, vehicle dynamics, nominal controller configuration, and historical context. Formally, the state vector is structured as:

$$\mathbf{s}(t) = \left[\mathbf{e}_{\text{track}}(t),\quad \mathbf{v}_{\text{dyn}}(t),\quad \kappa_{\text{path}}(t),\quad \mathbf{a}_{\text{base}}(t),\quad \mathrm{H}_M(t)\right]^T \tag{12}$$

where $\mathbf{e}_{\text{track}}(t) = [e_x(t), e_y(t), e_\psi(t)]$ contains the longitudinal, lateral, and heading tracking errors; $\mathbf{v}_{\text{dyn}}(t)$ comprises the ego vehicle's longitudinal velocity $v(t)$, angular velocity $\omega(t)$, the leader

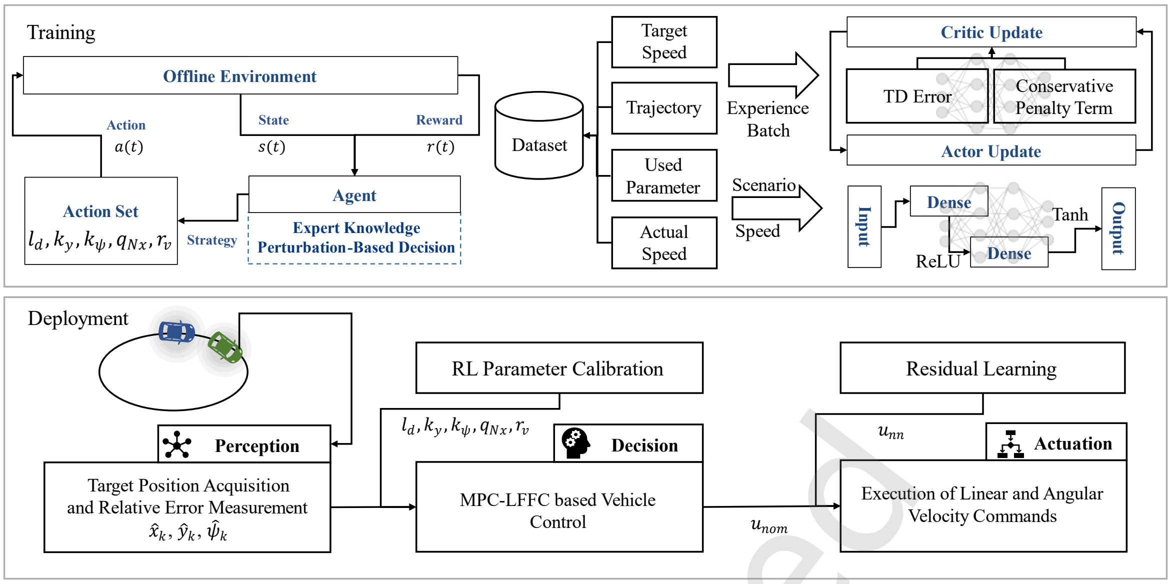

Figure 3: Data-driven improvement module for MPC-LFFC control.

vehicle velocity $v_{\text{leader}}(t)$, and the ego vehicle's acceleration $a(t)$ and jerk $j(t)$ (i.e., $\mathbf{v}_{\text{dyn}}(t)=[v(t),\omega(t),v_{\text{leader}}(t),a(t),j(t)]^T$; $\kappa_{\text{path}}(t)=[\kappa(t),\kappa_{\text{ahead}}(t)]$ encapsulates the local path curvature and preview geometry; $\mathbf{a}_{\text{base}}(t)$ denotes the expert-designed nominal controller parameter configuration; and $\mathrm{H}_M(t)=\{\mathbf{s}(\tau),\mathbf{a}(\tau)\}_{\tau=t-M}^{t-1}$ is a historical context window of length M introduced to mitigate partial observability.

The action space consists of five continuous controller parameters:

$$\mathbf{a}(t)=\left[l_d(t),\quad k_y(t),\quad k_\psi(t),\quad q_{N,x}(t),\quad r_v(t)\right]^T \tag{13}$$

where l_d denotes the preview distance used by the path-tracking module to select the look-ahead point, k_y and k_ψ are the lateral LFFC feedback gains, and $q_{N,x}$ and r_v are the terminal-state and control-effort weights in the longitudinal MPC controller.

Since online trial-and-error is prohibitively expensive in real-world scenarios, we adopt an offline training paradigm. As illustrated in Fig. 3, We first obtain a reasonably effective expert policy through sensitivity analysis of each parameter with respect to vehicle speed. The training dataset is then collected by sampling parameter perturbations around this expert configuration. Let τ denote a delay offset and H the evaluation horizon. The reward function is defined as:

$$r(t)=-\left(J_{\text{track}}+J_{\text{comfort}}+J_{\text{rate}}+J_{\text{safety}}\right) \tag{14}$$

where the components are designed as follows:

1. **Tracking Performance:** The future tracking cost is evaluated over a shifted horizon to associate the current parameter choice with its delayed consequences:

$$J_{\text{track}}=\frac{1}{H}\sum_{i=0}^{H-1}\left[w_x e_x^2(t+\tau+i)+w_y e_y^2(t+\tau+i)+w_\psi e_\psi^2(t+\tau+i)\right] \tag{15}$$

2. **Ride Comfort:** Aggressive control actions are penalized via acceleration and jerk profiles:

$$J_{\text{comfort}}=w_a a^2(t)+w_j j^2(t) \tag{16}$$

3. **Action Smoothness:** Abrupt parameter fluctuations between consecutive steps are suppressed via an action-rate penalty:

$$J_{\text{rate}}=w_r\left\|\frac{\mathbf{a}(t)-\mathbf{a}(t-1)}{\Delta\mathbf{a}}\right\|_2^2 \tag{17}$$

where $\Delta\mathbf{a}$ represents the allowable adjustment bounds for normalization.

4. **Safety Constraint:** A large constant penalty $J_{\text{safety}}=C_{\text{safe}}$ is imposed whenever the underlying safety guard is activated, discouraging out-of-bound parameter recommendations.

The parameter calibration policy is trained using CQL within an offline Actor-Critic framework. CQL mitigates value overestimation caused by distributional shift, where the policy queries out-of-distribution actions not covered in the static dataset D.

In the Critic learning phase, the Q-network $Q_\theta(\mathbf{s},\mathbf{a})$ is optimized by minimizing a conservative objective. Since the action space is continuous, we approximate the integral over actions by sampling K actions from a uniform distribution or from the current policy. The practical CQL loss is:

$$\begin{aligned}\min_\theta L_{\text{CQL}}(\theta)=\alpha_{\text{cql}}\cdot\mathrm{E}_{\mathbf{s}\sim\mathrm{D}}\left[\log\left(\frac{1}{K}\sum_{i=1}^{K}\exp\left(Q_\theta(\mathbf{s},\mathbf{a}_i)\right)\right)-\mathrm{E}_{\mathbf{a}\sim\mathrm{D}}\left[Q_\theta(\mathbf{s},\mathbf{a})\right]\right]\\+\frac{1}{2}\mathrm{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathrm{D}}\left[\left(Q_\theta(\mathbf{s},\mathbf{a})-\mathrm{B}^\pi Q_{\bar\theta}(\mathbf{s},\mathbf{a})\right)^2\right]\end{aligned} \tag{18}$$

where α_{cql} is a trade-off coefficient, $Q_{\bar\theta}$ is the target Q-network, B^π is the empirical Bellman operator defined as $\mathrm{B}^\pi Q_\theta(\mathbf{s},\mathbf{a})=r+\gamma\mathrm{E}_{\mathbf{a}'\sim\pi_\phi(\cdot|\mathbf{s}')}[Q_\theta(\mathbf{s}',\mathbf{a}')]$, and $\{\mathbf{a}_i\}_{i=1}^K$ are samples drawn from a proposal distribution (e.g., uniform over the action space or from the current policy) to approximate the log-sum-exp term.

In the Actor phase, the policy $\pi_\phi(\mathbf{a}|\mathbf{s})$ is updated by maximizing the expected value under the conservative critic, regularized by an entropy term:

$$\max_\phi L_{\text{Actor}}(\phi)=\mathrm{E}_{\mathbf{s}\sim\mathrm{D},\mathbf{a}\sim\pi_\phi(\cdot|\mathbf{s})}\left[Q_\theta(\mathbf{s},\mathbf{a})\right]+\beta\cdot\mathrm{H}\left(\pi_\phi(\cdot|\mathbf{s})\right) \tag{19}$$

where β determines the entropy regularization weight.

5.2. Residual learning compensation via neural networks

However, even with optimally tuned parameters, the model-based controller remains subject to unmodeled nonlinearities, such as actuator delays, tire slip, motor dead zones, and battery-voltage-dependent dynamics. These effects cause the actual vehicle motion to deviate from the commanded state. To compensate for these residual discrepancies, the second level introduces a neural-network-based residual learning module that predicts feedforward corrections to the nominal control commands. This two-tiered approach ensures that the advantages of the model-based controller are retained while its performance is continuously improved through data-driven adaptation. The final command applied to the vehicle is expressed as

$$\begin{aligned} v_{\text{cmd}}(t) &= v_{\text{nom}}(t) + \Delta v_{\text{nn}}(t) \\ \omega_{\text{cmd}}(t) &= \omega_{\text{nom}}(t) + \Delta \omega_{\text{nn}}(t) \end{aligned} \tag{20}$$

where $v_{\text{nom}}(t)$ and $\omega_{\text{nom}}(t)$ are the nominal commands generated by the MPC-LFFC controller, and $\Delta v_{\text{nn}}(t)$ and $\Delta \omega_{\text{nn}}(t)$ are the residual compensation terms predicted by the neural network.

The network input comprises the current tracking state, its temporal evolution, and actuator-related operating conditions. Specifically, the feature vector includes the tracking errors (e_x, e_y, e_ψ), their discrete-time derivatives, and a set of historical statistics (moving averages and lagged observations) computed over a sliding time window. This design enables the network to distinguish persistent dynamic trends from measurement noise. In addition, the nominal commands $(v_{\text{nom}}, \omega_{\text{nom}})$, measured velocities (v, ω), and battery voltage are incorporated to reflect the current operating condition and actuator capability. To account for actuation latency, the network is trained with a time-shifted label construction strategy. Let τ_d denote the estimated delay in terms of control steps (an integer). Instead of using the instantaneous execution error, the target residual is defined as

$$\mathbf{u}_{\text{label}}(t) = \begin{bmatrix} v_{\text{nom}}(t) - v(t + \tau_d \Delta t) \\ \omega_{\text{nom}}(t) - \omega(t + \tau_d \Delta t) \end{bmatrix} \tag{21}$$

where $t + \tau_d \Delta t$ denotes the future time instant after the delay. Thus, the network input at time t is associated with the execution discrepancy observed at a future instant, enabling the network to learn predictive compensation that proactively mitigates the effects of actuation delays.

As shown in Fig. 3, the residual predictor is implemented as a multilayer perceptron with Batch Normalization. Training is performed by minimizing a composite loss function,

$$\begin{aligned} \mathrm{L} &= \mathrm{L}_{\text{MSE}} + \lambda_{\text{smooth}} \mathrm{L}_{\text{smooth}} \\ &= \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{u}_{\text{label},i} - \left[\Delta v_{\text{nn},i}, \Delta \omega_{\text{nn},i} \right]^T \right\|_2^2 \\ &+ \frac{\lambda_{\text{smooth}}}{N-1} \sum_{i=2}^{N} \left\| \left[\Delta v_{\text{nn},i}, \Delta \omega_{\text{nn},i} \right]^T - \left[\Delta v_{\text{nn},i-1}, \Delta \omega_{\text{nn},i-1} \right]^T \right\|_2^2 \end{aligned} \tag{22}$$

The mean-squared-error term minimizes the discrepancy between the predicted and target residuals, while the smoothness term penalizes abrupt variations between consecutive predictions to reduce high-frequency oscillations and promote control continuity. Gradient norm clipping is applied during optimization to further improve training stability.

During online deployment, the residual network operates together with the model-based controller, subject to additional safety constraints: the compensation output is disabled when the vehicle is nearly stationary, when tracking errors exceed predefined thresholds, or when the battery voltage falls outside the normal operating range. Furthermore, compensation commands that conflict with the desired error-reduction direction are suppressed. Finally, rate limits and saturation constraints are enforced on both the residual output and the aggregated control command to ensure compliance with actuator capabilities.

6 Experiments

6.1. Experiments setup

To verify the feasibility of the proposed platoon tracking control method, a two-vehicle platoon scenario is first established in the Gazebo simulation environment. The simulation system consists of a leading vehicle, a following vehicle, a reference path, and a low-level velocity control interface. It is used to verify the correctness of the reference-point construction, platoon motion logic, and control command generation before real-vehicle experiments. The simulation scenario and the platoon tracking performance of the two vehicles are shown in Fig. 4.

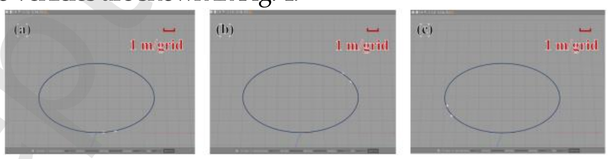

Figure 4: Simulation scenario and platoon tracking performance.

The real-vehicle experiments are conducted using two Robot Operating System (ROS) based wheeled mobile platform. The experimental vehicles and their sensor configurations are shown in Fig. 5. The vehicles are equipped with wheel odometry, an IMU and a GPS receiver to obtain vehicle velocity, heading angle, and global position observations. Based on the fused vehicle state, the leading-vehicle state, and the reference path information, the controller computes the linear and angular velocity commands online and sends them to the low-level actuator of the following vehicle.

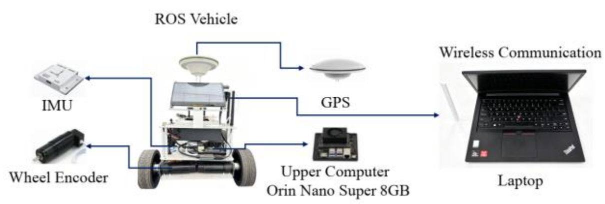

Figure 5: Experimental vehicles and sensor configurations.

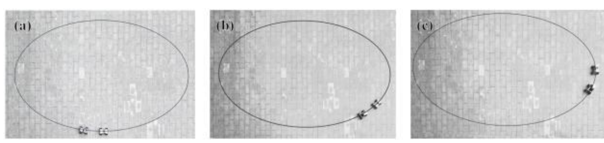

Figure 6: Real-world test site.

The real-world test site is shown in Fig. 6. The experimental path is set as a closed elliptical trajectory, with the major and minor axes equal to $5\,\mathrm{m}$ and $3\,\mathrm{m}$, respectively. The desired platoon distance is set to $1\,\mathrm{m}$. To evaluate the adaptability of the control method under different velocity conditions, a variable-speed operating condition is adopted. The reference speed of the leading vehicle is gradually increased from 0.4m/s to 1.2m/s with an interval of 0.1m/s , and each speed segment lasts for 90s . A smooth transition is applied during speed switching to reduce the influence of abrupt velocity changes on the closed-loop response.

The reference speed profile over time is shown in Fig. 7. Unless otherwise specified, all experiments use the same reference path, speed schedule, and initial platoon configuration.

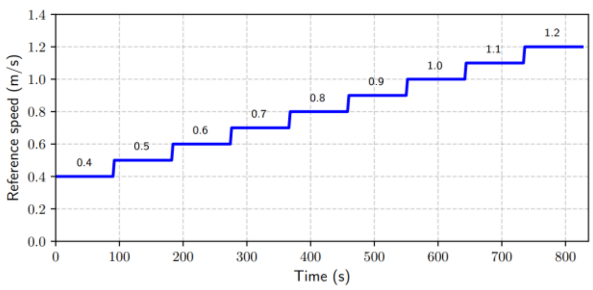

Figure 7: Reference speed profile over time.

The evaluation metrics include the position tracking error and the velocity tracking error. Let the desired tracking point and the actual position of the following vehicle at the k-th sampling instant be denoted as $\mathbf{p}_{r,k}=\left[x_{r,k},y_{r,k}\right]^T$ and $\mathbf{p}_{f,k}=\left[x_{f,k},y_{f,k}\right]^T$, respectively. The Euclidean position error is defined as

$$e_{p,k}=\|\mathbf{p}_{r,k}-\mathbf{p}_{f,k}\|_2 \tag{23}$$

The position error is further projected onto the body-fixed coordinate frame of the following vehicle, yielding the longitudinal error $e_{x,k}$ and lateral error $e_{y,k}$, which are used to evaluate the tracking performance in the forward and lateral directions, respectively. The velocity tracking errors are defined as the differences between the controller outputs and the vehicle feedback velocities:

$$\begin{aligned} e_{v,k} &= v_{\mathrm{nom},k}-v_{\mathrm{cur},k} \\ e_{\omega,k} &= \omega_{\mathrm{nom},k}-\omega_{\mathrm{cur},k} \end{aligned} \tag{24}$$

where $v_{\mathrm{cur},k}$ and $\omega_{\mathrm{cur},k}$ are the actual linear and angular velocities fed back by the vehicle, respectively. The mean absolute error (MAE), root mean square error (RMSE), and maximum absolute error (MAX) are used to statistically evaluate the above error sequences, thereby providing a comprehensive assessment of the tracking accuracy, fluctuation level, and extreme error magnitude of different methods.

6.2. Effectiveness of the data fusion module

Since our current platform lacks an independent high-precision positioning reference (e.g., RTK-GNSS or motion capture), the position-related errors reported in Table 1 are tracking errors computed within each method's own state estimate and are presented for descriptive purposes. we also report velocity errors, which are derived from wheel odometry and IMU feedback and do not depend on any position estimate. All experiments are measured during the same day.

To verify the effectiveness of the proposed data fusion module, we conduct ablation experiments with the following baselines for comparison:

- **ODOM + IMU:** The tracking error terms used in MPC control are computed solely from odometry and IMU outputs.
- **GPS only:** The tracking error is computed using GPS measurements alone.

Table 1: Comparison of tracking errors across different localization sources.

Metric RMSE	GPS only	ODOM + IMU	Proposed fusion
		Position-related errors	
e_x	0.3663 $\pm$ 0.2078	0.2830 $\pm$ 0.0439	**0.0675 $\pm$ 0.0167**
e_y	0.1954 $\pm$ 0.0943	0.2582 $\pm$ 0.0319	**0.0585 $\pm$ 0.0008**
e_p	0.4156 $\pm$ 0.2030	0.3843 $\pm$ 0.0617	**0.0899 $\pm$ 0.0149**
		Velocity errors	
e_v	0.0461 $\pm$ 0.0046	0.0333 $\pm$ 0.0007	**0.0295 $\pm$ 0.0013**
e_ω	0.1880 $\pm$ 0.0019	0.1955 $\pm$ 0.0028	**0.1877 $\pm$ 0.0023**

Although the position-related errors in Table 1 should not be directly compared across methods, the proposed fusion method consistently yields the smallest error magnitudes in both longitudinal and lateral directions within its own state-estimation framework. For cross-method comparison, we focus on the velocity tracking errors: the proposed fusion method achieves the lowest linear velocity RMSE (0.0295 m/s), outperforming ODOM+IMU (0.0333 m/s) and GPS-only (0.0461 m/s), with comparable angular velocity performance (0.1877 rad/s vs. 0.1880 rad/s for GPS-only). These results indicate that the proposed fusion method provides more stable state estimates for the controller, particularly in the longitudinal direction where platoon tracking accuracy is most critical.

Fig. 8 further illustrates the trajectories of the three localization configurations. In Fig. 8(a), the ODOM + IMU baseline produces a smooth trajectory; however, due to the absence of GPS correction, the path exhibits systematic drift and global offset over time. In Fig. 8(b), the GPS-only baseline is visibly affected by signal fluctuations that result in large tracking errors. In contrast, the proposed fusion method in Fig. 8(c) maintains the highest positional accuracy, combining smooth local propagation with globally bounded error.

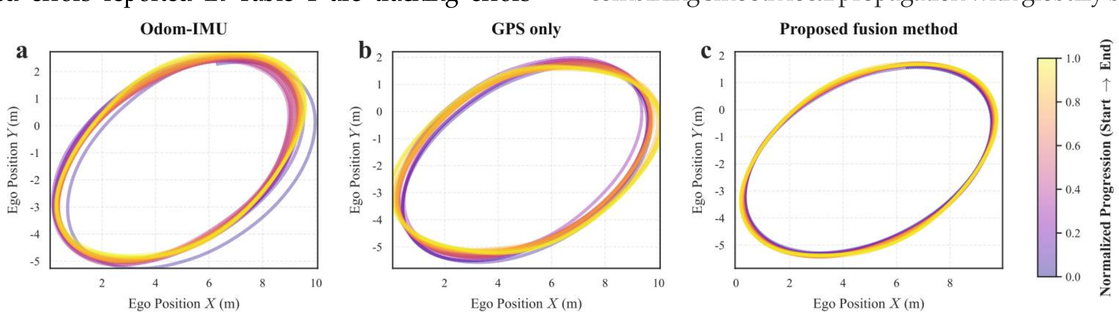

Figure 8: Trajectory comparison of the three localization methods, visualized via error ellipses.

6.3. *Effectiveness of MPC-LFFC model and parameter calibration*

To evaluate the effectiveness of the proposed MPC-LFFC controller and CQL parameter calibration strategy, we compare five configurations below:

- **Traditional MPC:** MPC with a standard lateral controller that computes the look-ahead error from lateral deviation and heading error, and applies proportional feedback for correction, without the proposed LFFC module.
- **Fixed Parameter:** MPC-LFFC with a single fixed parameter set across all operating conditions.
- **Expert Experience:** MPC-LFFC with manually designed speed-adaptive parameters, as listed in Table 2.
- **Normal RL:** MPC-LFFC with parameters calibrated by a standard offline RL agent trained without a conservative loss function.
- **Proposed CQL:** MPC-LFFC with parameters calibrated by the proposed offline CQL agent.

Note that all experiments are measured in the same day.

Table 2: Expert-designed speed-adaptive parameter table for MPC-LFFC

v_{ref}	l_d	k_y	k_ψ	$q_{N,x}$	r_v
0.4	0.55	2.20	1.90	15.0	0.70
0.5	0.65	2.00	1.80	16.0	0.80
0.6	0.80	1.80	1.60	17.0	0.90
0.7	0.95	1.65	1.45	18.0	1.00
0.8	1.10	1.50	1.30	20.0	1.15
0.9	1.25	1.35	1.20	22.0	1.30
1.0	1.40	1.25	1.10	24.0	1.50
1.1	1.55	1.15	1.00	26.0	1.70
1.2	1.70	1.05	0.95	28.0	2.00

As shown in Table 3, among the baselines, Traditional MPC without the proposed LFFC module yields the largest errors across all metrics, highlighting the necessity of the proposed lateral-error-constrained structure. Fixed Parameter, despite having the same controller model but with static parameters, exhibits the smallest longitudinal standard deviation but poor MAE, indicating stable but biased behavior. Expert Experience improves upon Fixed Parameter by adapting parameters to vehicle speed, yet relies on manual heuristics and cannot fully generalize. Normal RL, trained without conservative regularization, suffers from distributional shift in offline data, resulting in degraded performance compared to Expert Experience in most metrics. In contrast, the proposed CQL agent learns robust parameter policies from historical logs while avoiding overestimation of out-of-distribution actions, achieving the best overall trade-off and superior accuracy in longitudinal, position, and velocity errors, with competitive lateral performance.

Fig. 9 visualizes the error distributions of the four parameter-calibration methods via trajectory error ellipses. The proposed CQL method exhibits the most concentrated ellipse, indicating more consistent tracking behavior across varying speed profiles. The Fixed Parameter, Expert Experience and Normal RL baselines produce noticeably larger ellipses, reflecting their sensitivity to operating condition changes.

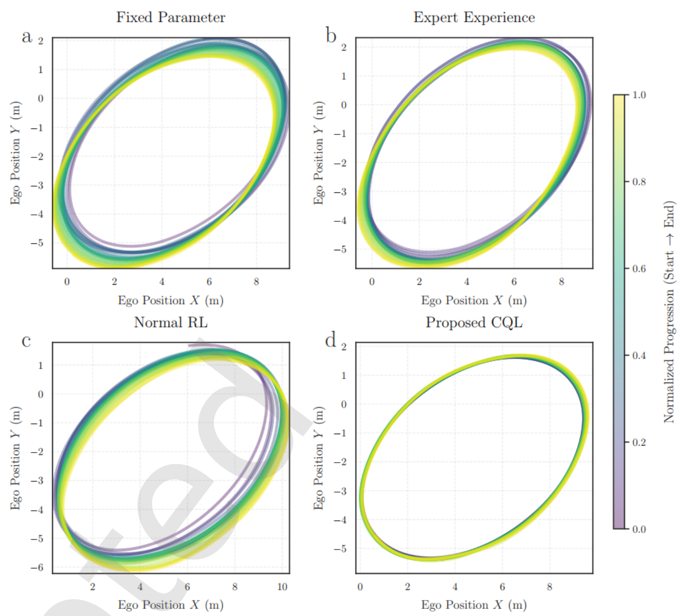

Figure 9: Trajectory error ellipse comparison of the four parameter calibration methods.

To further analyze the relationship between optimal parameters and vehicle speed, Fig. 10 illustrates the distribution of calibrated parameter values across different reference velocities, derived from the CQL training. The median trends of all five parameters exhibit clear speed-dependent correlations: the lookahead distance l_d increases with speed, the lateral gains k_y and k_ψ decrease, and the MPC weights $q_{N,x}$ and r_v grow nearly monotonically. However, the learned distributions exhibit non-negligible variance at each speed level, as evidenced by the interquartile ranges and whiskers in all subplots. This indicates that the optimal parameters are not solely determined by vehicle speed, but are also influenced by additional contextual factors such as path curvature, tracking error history, and instantaneous dynamic conditions. Unlike the expert experience, which provides a fixed parameter for each speed bin, the CQL agent learns to modulate parameters conditioned on the full state observation $\mathbf{s}(t)$, enabling finer-grained adaptation to the actual driving context.

6.4. *Effectiveness of the residual learning module*

To quantitatively analyze the effectiveness of the residual learning module, comparative experiments are conducted under a target velocity condition of $v = 0.6\mathrm{m/s}$ across both offline validation datasets and closed-loop online deployment.

Table 3: Comparison of tracking performance metrics.

Metric	Traditional MPC	Fixed Parameter	Expert Experience	Normal RL	Proposed CQL
		Longitudinal error e_x [m]			
μ	0.1389	0.0920	0.0709	0.0842	**0.0502**
σ	0.0664	**0.0120**	0.0182	0.0244	0.0236
RMSE	0.1541	0.0928	0.0741	0.0896	**0.0554**
MAE	0.1390	0.0920	0.0716	0.0879	**0.0507**

Lateral error e_y [m]					
μ	-0.0964	-0.0359	-0.0334	-0.0331	**-0.0255**
σ	0.1343	**0.0327**	0.0444	0.0436	0.0487
RMSE	0.1653	**0.0485**	0.0556	0.0549	0.0550
MAE	0.1259	0.0391	0.0415	0.0416	**0.0387**
Position error $e_p = \sqrt{e_x^2 + e_y^2}$ [m]					
μ	0.1967	0.1032	0.0892	0.1017	**0.0711**
σ	0.1119	**0.0178**	0.0268	0.0260	0.0322
RMSE	0.2264	0.1047	0.0935	0.1051	**0.0781**
MAE	0.1967	0.1032	0.0892	0.1017	**0.0711**
Velocity error e_v [m/s]					
μ	0.0272	0.0162	0.0171	0.0147	**0.0104**
σ	0.0553	0.1808	0.1825	0.1896	**0.0336**
RMSE	0.0617	0.1815	0.1833	0.1902	**0.0352**
MAE	0.0413	0.1644	0.1645	0.1684	**0.0227**

On one hand, offline evaluation is performed on an uncompensated real-world verification dataset. The features are fed into the trained network to directly output the feedforward compensation terms, which are then compared with the uncompensated MPC-LFFC dataset. As summarized in Table 4, the compensation achieves a direction alignment rate of $V_{\text{Align}} = 80.92\%$ for linear velocity and $W_{\text{Align}} = 71.44\%$ for angular velocity toward the immediate future frame.

In terms of physical tracking errors, the default MPC-LFFC controller yields an expected future linear velocity tracking error of 0.0128m/s, which drops to 0.0077m/s upon applying the compensation, representing an expected longitudinal tracking improvement of 40.03%. Similarly, the future angular velocity error is reduced from 0.0376rad/s to 0.0320rad/s, delivering a 15.01% improvement in lateral heading accuracy. These results substantiate that the network successfully learns the inverted kinematic discrepancy from historical logs.

Table 4 Offline predictive evaluation of residual learning.

Metric	Default MPC	With RRL	Improvement
Future Linear Error (m/s)	0.0128	0.0077	40.03%
Future Angular Error (rad/s)	0.0376	0.0320	15.01%
Linear Direction Alignment V_{Align}		80.92%	
Angular Direction Alignment W_{Align}		71.44%	

On the other hand, to validate the closed-loop stability and real-time performance, online physical tests were carried out on the same day under identical environmental setups. The tracking discrepancies between the actual vehicle velocities and the reference commands generated by the MPC-LFFC (v_{nom}, ω_{nom}) are thoroughly evaluated. The collected dataset comprises 49780 frames for the testing profile.

Table 5: Online tracking performance comparison for residual learning module.

Dimension	Index	W/O Comp.	With Comp.	Improvement
Linear Velocity v	RMSE (m/s)	0.0230	0.0160	30.66%
	MAE (m/s)	0.0154	0.0117	24.06%
	MAX (m/s)	0.2847	0.1051	63.09%
Angular Velocity ω	RMSE (rad/s)	0.0446	0.0432	3.21%
	MAE (rad/s)	0.0350	0.0341	2.67%
	MAX (rad/s)	0.3263	0.1867	42.79%

Note: W/O Comp.: Without Compensation; Comp.: With Compensation.

The detailed statistical evaluation is presented in Table 5. The residual feedforward compensation yields the most significant improvements in longitudinal velocity tracking, reducing RMSE and MAE by 30.66% and 24.06%, respectively, while the maximum absolute error drops by 63.09%, effectively suppressing transient overshoots. For lateral angular velocity tracking, the improvements are more moderate, with RMSE and MAE improving by 3.21% and 2.67%, respectively, while the maximum boundary error is reduced by 42.79%, enhancing cornering safety and preventing aggressive oscillations.

6.5 String Stability Validation

Since the real-world experiments are limited to two vehicles and thus cannot directly evaluate disturbance propagation along a vehicle string, we complement the physical experiments with a multi-vehicle simulation study to assess string stability. The simulated platoon consists of one leader and three followers. The desired adjacent spacing is set to $d_{des} = 1.0\,\text{m}$. To excite the platoon dynamics, the leader follows a periodic velocity profile:

$$v_L(t) = 0.8 + 0.4\sin(2\pi f t), \qquad f = 0.02\ \text{Hz}, \tag{25}$$

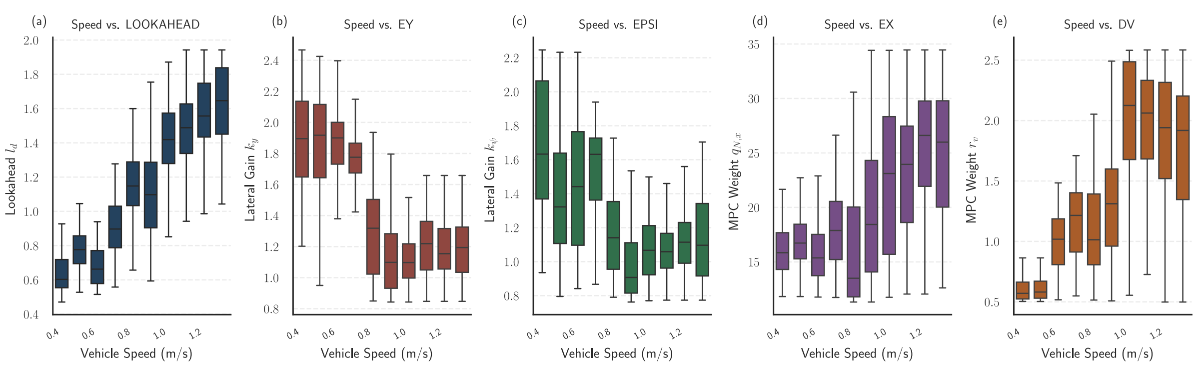

Figure 10: Learned MPC-LFFC parameters as a function of reference velocity.

where $v_L(t)$ denotes the leader velocity command and f is the excitation frequency. Therefore, the leader speed varies periodically between 0.4 m/s and 1.2 m/s. Random actuation uncertainties are injected into the follower velocity commands.

To reduce the influence of randomness from a single simulation run, we repeated the experiment five times. For each run, we evaluated the velocity disturbance amplification ratio:

$$A_{v,i} = \frac{\mathrm{std}(v_i)}{\mathrm{std}(v_L)}, \tag{26}$$

where v_i is the velocity of the i-th vehicle and $\mathrm{std}(\cdot)$ denotes the standard deviation over the analyzed time interval. A value of $A_{v,i} < 1$ indicates that the velocity fluctuation of vehicle i is smaller than that of the leader.

We also evaluated the adjacent spacing error:

$$e_{ij}(t) = d_{ij}(t) - d_{\mathrm{des}}, \tag{27}$$

where $d_{ij}(t)$ is the actual distance between two adjacent vehicles and d_{des} is the desired spacing. The spacing-error RMS amplification ratios are defined as:

$$A_{e,23} = \frac{\mathrm{RMS}(e_{23})}{\mathrm{RMS}(e_{12})}, \quad A_{e,34} = \frac{\mathrm{RMS}(e_{34})}{\mathrm{RMS}(e_{23})}, \tag{28}$$

where $\mathrm{RMS}(\cdot)$ denotes the root-mean-square value.

The statistical results over five repeated simulations are reported in Table 6. The velocity disturbance amplification ratios of the three followers are all below 1, and their 95% confidence interval upper bounds are also below 1. This indicates that the velocity disturbance is attenuated rather than amplified along the platoon.

For the spacing errors, the RMS amplification ratios remain close to or below unity. Specifically, $A_{e,23}$ is slightly above 1, while $A_{e,34}$ is below 1. This suggests that the adjacent spacing errors remain bounded and do not show a sustained increasing trend along the platoon. As shown in Fig. 11 and Fig.12, although this simulation study is not a formal theoretical proof, it provides simulation-based evidence that the proposed controller exhibits favorable string-stability behavior under the tested periodic speed excitation.

Table 6: Comparison of tracking performance metrics.

Metric	Mean	Std.	95%CI
$A_{v,2}$	0.9678	0.0041	[0.9627, 0.9729]
$A_{v,3}$	0.9654	0.0023	[0.9625, 0.9683]
$A_{v,4}$	0.9484	0.0055	[0.9416, 0.9552]
$A_{e,23}$	1.0196	0.0134	[1.0029, 1.0362]
$A_{e,34}$	0.9746	0.0114	[0.9605, 0.9887]

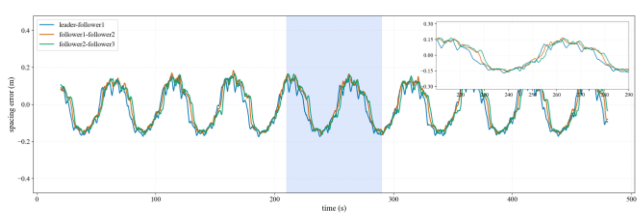

Figure 11: Velocity response of the platoon under periodic leader excitation.

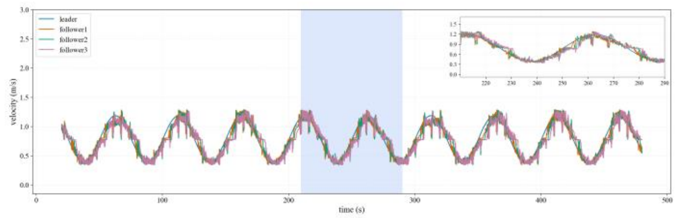

Figure 12: Adjacent inter-vehicle spacing error response under periodic leader excitation.

7 Conclusions

This paper presents a physics-and-data-combined framework for robust vehicle platooning that integrates multi-sensor fusion, MPC-LFFC control with offline CQL-based parameter calibration, and neural residual learning compensation. The fusion module combines wheel odometry, IMU, and GPS through a bounded prediction-correction scheme, achieving lower longitudinal, lateral, and position errors than single-source baselines. The CQL agent adapts controller parameters to varying speeds, outperforming fixed, expert, and standard RL configurations with the smallest longitudinal (0.0554 m) and position RMSE (0.0781 m). The residual learner further compensates for unmodeled actuator dynamics, reducing future linear velocity error by 40.03% and peak longitudinal overshoot by 63.09% in online tests.

A limitation of this study is that the fusion module lacks RTK-GNSS ground truth for absolute positioning evaluation. While the learning modules are constrained by bounded parameter ranges, amplitude limits, and safety guards, a formal stability proof remains open for future work. Future directions also include multi-vehicle cooperative platooning with heterogeneous dynamics and formation reconfiguration under varying communication topologies, generalization to adverse weather and low-friction surfaces, incorporation of communication delays and packet dropouts with delay-robust MPC-LFFC and residual modules, and online safe fine-calibration for changing vehicle conditions.

References

Alaba, S.Y., 2024. GPS-IMU sensor fusion for reliable autonomous vehicle position estimation. https://doi.org/10.48550/arXiv.2405.08119

Chen, Q., Wang, S., Gao, B., Zhan, Z., Zhong, R., 2025. Robust longitudinal and lateral control for mixed-vehicular platoons with string stability guarantees. Nonlinear Dyn. 113 (20), 27853–27875.

Cheng, J., Zhang, L., Chen, Q., Fu, Z., Du, L., 2024. High precision and robust vehicle localization algorithm with visual-LiDAR-IMU fusion. IEEE Trans. Veh. Technol. 73 (8), 11029–11043.

Cheng, Q., Zhang, Y., Gu, Z., Wang, Z., Liu, H., Lin, Y., et al., 2025. A hybrid physics-based and data-driven approach for car-following behavior modeling and analysis. Transp. Res. Part C Emerg. Technol. 177, 105207.

Chu, D., Li, H., Zhao, C., Zhou, T., 2022. Trajectory tracking of autonomous vehicle based on model predictive control with PID feedback. IEEE Trans. Intell. Transp. Syst. 24 (2), 2239–2250.

Dai, S., Li, S., Tang, H., Ning, X., Fang, F., Fu, Y., et al., 2024. MARP:

A cooperative multiagent DRL system for connected autonomous vehicle platooning. IEEE Internet Things J. 11 (20), 32454–32463.
Ding, H., Li, W., Xu, N., Zhang, J., 2022. An enhanced eco-driving strategy based on reinforcement learning for connected electric vehicles: cooperative velocity and lane-changing control. J. Intell. Connect. Veh. 5 (3), 316–332.
Farimani, M.J., Amirabadi, R.K., Nikkhouy, D., Abdolbaghi, M., Rastegarmoghaddam, M., Samadzadeh, S., et al., 2026. A systematic review and taxonomy of reinforcement learning-model predictive control integration for linear systems. https://doi.org/10.48550/arXiv.2604.21030
Feng, Y., Yu, S., Sheng, E., Li, Y., Shi, S., Yu, J., et al., 2023. Distributed MPC of vehicle platoons considering longitudinal and lateral coupling. IEEE Trans. Intell. Transp. Syst. 25 (3), 2293–2310.
Gao, F., Yang, B., Ning, C., Guan, X., 2025. Virtual leader-follower based platooning under mixed traffic: A data-driven distributionally robust MPC method. IEEE Trans. Veh. Technol. 74 (4), 5678–5692.
Gao, H., Yen, C.-C., Zhang, M., 2024. DRL based platooning control with traffic signal synchronization for delay and fuel optimization. Transp. Res. Part C Emerg. Technol. 163, 104655.
Gratzer, A.L., Broger, M.M., Schirrer, A., Jakubek, S., 2024. Two-layer MPC architecture for efficient mixed-integer-informed obstacle avoidance in real-time. IEEE Trans. Intell. Transp. Syst. 25 (10), 13767–13784.
Guo, J., Guo, H., Liu, J., Cao, D., Chen, H., 2022. Distributed data-driven predictive control for hybrid connected vehicle platoons with guaranteed robustness and string stability. IEEE Internet Things J. 9 (17), 16308–16321.
Huang, K., Shi, B., Li, X., Li, X., Huang, S., Li, Y., 2022. Multi-modal sensor fusion for auto driving perception: A survey. https://doi.org/10.48550/arXiv.2202.02703
Huo, J., Gu, Z., Liu, Z., Wang, S., Laporte, G., 2025. A heteroscedastic robust Bayesian optimization method for solving simulation-based transportation problems. Transp. Sci. 59 (6), 1353–1374.
Jiang, M., Chu, Z., Pan, Y., Yuan, H., Han, Q., Zou, J., et al., 2026. Multi-sensor fusion on autonomous driving perception: A comprehensive survey. Multimodal Transp. 5, 100325.
Joubert, N., Reid, T.G., Noble, F., 2020. Developments in modern GNSS and its impact on autonomous vehicle architectures. In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), 2029–2036.
Lai, X., Yang, Z., Xie, J., Liu, Y., 2024. Reinforcement learning in transportation research: Frontiers and future directions. Multimodal Transp. 3 (4), 100164.
Li, Q., Zhang, P., Yao, H., Chen, Z., Li, X., 2024a. Online learning-based model predictive trajectory control for connected and autonomous vehicles: modeling and physical tests. J. Intell. Connect. Veh. 7, 86–96.
Li, W., Rios-Torres, J., Wang, B., Khattak, Z.H., 2024b. Experimental assessment of communication delay's impact on connected automated vehicle speed volatility and energy consumption. Commun Transp Res 4, 100136.
Li, Z., Zhao, P., Jiang, C., Huang, W., Liang, H., 2022. A learning-based model predictive trajectory planning controller for automated driving in unstructured dynamic environments. IEEE Trans. Veh. Technol. 71 (6), 5944–5959.
Lin, P., Chen, Z., Pei, M., Ding, Y., Qu, X., Zhong, L., 2024. Multiple emergency vehicle priority in a connected vehicle environment: A cooperative method. IEEE Trans. Intell. Transp. Syst. 25 (1), 173–188.
Liu, X., Wen, W., Hsu, L.-T., 2023a. GLIO: Tightly-coupled GNSS/LiDAR/IMU integration for continuous and drift-free state estimation of intelligent vehicles in urban areas. IEEE Trans. Intell. Veh. 9 (1), 1412–1422.
Liu, Z., Lyu, C., Wang, Z., Wang, S., Liu, P., Meng, Q., 2023b. A Gaussian-process-based data-driven traffic flow model and its application in road capacity analysis. IEEE Trans. Intell. Transp. Syst. 24 (2), 1544–1563.
Liu, Z., Zhou, Z., Gu, Z., Liu, S., Liu, P., Zhang, Y., et al., 2025. TRIP: Transport reasoning with intelligence progression—A foundation framework. Transp. Res. Part C Emerg. Technol. 179, 105260.
Long, K., Liang, Z., Shi, H., Shi, L., Chen, S., Li, X., 2024. Traffic oscillation mitigation with physics-enhanced residual learning (PERL)-based predictive control. Commun Transp Res 4, 100154.
Ma, C., Zhou, H., Zhang, P., Ma, K., Shi, H., Li, X., 2025. Safety assurance adaptive control for modular autonomous vehicles. Commun Transp Res 5, 100204.
Ma, F., Wang, J., Zhu, S., Gelbal, S.Y., Yang, Y., Aksun-Guvenc, B., et al., 2020. Distributed control of cooperative vehicular platoon with nonideal communication condition. IEEE Trans. Veh. Technol. 69 (8), 8207–8220.
Ma, Y.-S., Che, W.-W., Deng, C., Wu, Z.-G., 2024. Data-driven distributed vehicle platoon control for heterogeneous nonlinear vehicle systems. IEEE Trans. Intell. Transp. Syst. 25 (3), 2373–2381.
Ping, P., Zhang, X., Tao, L., Shi, Q., Tian, Y., Yan, J., et al., 2025. A comprehensive survey on multi-sensor information processing and fusion for BEV perception in autonomous vehicles. Inf. Fusion 126, 103653.
Rebelo, M., Rafael, S., Bandeira, J.M., 2024. Vehicle platooning: A detailed literature review on environmental impacts and future research directions. Future Transp. 4 (2), 591–607.
Shi, Q., Zhang, H., 2022. Learning-based H∞ path-following controller design for autonomous ground vehicles subject to stochastic delays and actuator constraints. IEEE Trans. Ind. Electron. 70 (4), 4138–4147.
Tian, B., Guo, Q., Xu, Z., Wang, M., Qu, X., 2026a. MPC-based controller switching strategy for string stability and safety of vehicle platoons considering communication delays boundary. IEEE Trans. Veh. Technol. 75 (1), 123–138.
Tian, K., Wen, Y., Wang, S., Huang, Y., Huang, L., 2026b. Physics-informed neural network model for trajectory tracking control of unmanned surface vehicles. IEEE Internet Things J. 13 (8), 16924–16938.
Trumpp, R., Hoornaert, D., Caccamo, M., 2023. Residual policy learning for vehicle control of autonomous racing cars. In:

Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), 1–6.

Vinoth, K., Sasikumar, P., 2024. Multi-sensor fusion and segmentation for autonomous vehicle multi-object tracking using deep Q networks. Sci. Rep. 14 (1), 31130.

Wang, H., Yuan, Y., Yang, X.T., Zhao, T., Liu, Y., 2023a. Deep Q learning-based traffic signal control algorithms: Model development and evaluation with field data. J. Intell. Transp. Syst. 27 (3), 314–334.

Wang, K., Shen, C., Li, X., Lu, J., 2025a. Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications. IEEE Trans. Intell. Transp. Syst. 26 (3), 2880–2896.

Wang, K., Wang, Y., Liu, B., Chen, J., 2023b. Quantification of uncertainty and its applications to complex domain for autonomous vehicles perception system. IEEE Trans. Instrum. Meas. 72, 1–17.

Wang, Z., Cheng, Q., Gu, Z., Liu, C., Cun, D., Shi, X., et al., 2025b. Analyzing mega-mobility systems in smart cities: A macro-micro integration with feedback paradigm empowered by artificial intelligence. Research 8, 0982.

Wang, Z., Lin, Y., Liu, Z., Dong, Y., Zheng, Y., Liu, P., et al., 2026. Traffic dynamics modeling with an extended S3 car-following model. Transp. Res. Part C Emerg. Technol. 183, 105494.

Wang, Z., Liu, Z., Lin, Y., Zhang, Y., Cheng, Q., 2025c. Day-to-day traffic flow dynamics with mixed autonomy considering link-level penetration rate evolution of autonomous vehicles. Proc. IEEE, 1–20. https://doi.org/10.1109/JPROC.2025.3562946

Wu, X., Min, H., Mao, Z., Yan, Y., Liu, Y., Wu, G., 2026. Prior-knowledge-guided model-based reinforcement learning for integrated longitudinal-lateral control of vehicular platoons. J. Intell. Connect. Veh. 9 (1), 1–15.

Xie, C., Zhang, Z., Su, A., Wu, B., 2024. Intelligent vehicle platooning transit. Commun Transp Res 4 (4), 100145.

Yang, K., Tang, X., Li, J., Wang, H., Zhong, G., Chen, J., et al., 2023. Uncertainties in onboard algorithms for autonomous vehicles: Challenges, mitigation, and perspectives. IEEE Trans. Intell. Transp. Syst. 24 (9), 8963–8987.

Zhang, H., Du, L., 2023. Platoon-centered control for eco-driving at signalized intersection built upon hybrid MPC system, online learning and distributed optimization. Part I: Modeling and solution algorithm design. Transp. Res. Part B Methodol. 172, 174–198.

Zhang, J., Sheng, L., Zhou, D., 2024. Path-guided formation-containment control for networked heterogeneous multi-vehicles systems. IEEE Trans. Intell. Transp. Syst. 25 (11), 18292–18304.

Zhang, L., Chen, F., Ma, X., Pan, X., 2020. Fuel economy in truck platooning: A literature overview and directions for future research. J. Adv. Transp. 2020 (1), 2604012.

Zhang, P., Huang, H., Zhou, H., Shi, H., Long, K., Li, X., 2025. Online adaptive platoon control for connected and automated vehicles via physics enhanced residual learning. Transp. Res. Part C Emerg. Technol. 178, 105242.

Zhang, Y., Qu, X., Tong, L., 2022. Optimal eco-driving control of autonomous and electric trucks in adaptation to highway topography: Energy minimization and battery life extension. IEEE Trans. Transp. Electrific. 8 (2), 2149–2163.

Zhang, Y., Wang, Z., Pan, Y., Ma, X., Zhang, C., Liu, Z., et al., 2026. PFM-TMLC: Physics-guided microscopic trajectory reconstruction from fixed sensors under emergencies. Transp. Res. Part E Logist. Transp. Rev. 213, 105014.

Zuo, Z., Yang, K., Wang, H., Wang, Y., Wu, Y., 2025. Distributed MPC for automated vehicle platoon: A path-coupled extended look-ahead approach. IEEE Trans. Intell. Veh. 10 (4), 2441–2453.

Author contributions

Chengqi Liu: Conceptualization, Methodology, Validation, Visualization, Writing Original Draft; **Qiang Ma**: Methodology, Validation, Formal Analysis, Data Curation, Writing Review & Editing; **Xiwu Wang**: Investigation, Data Curation, Writing Review & Editing; **Qiang Sun**: Investigation, Writing Review & Editing; **Yinke Sun**: Investigation, Writing Review & Editing; **Zhiyuan Liu**: Supervision, Writing Review & Editing; **Nan Zheng**: Writing Review & Editing; **Kai Huang**: Supervision, Conceptualization, Funding Acquisition, Writing Review & Editing.

Replication and data sharing

The data and codes used in this study have been deposited to the ETS-Data repository (Dataset ID: ETS-Data-26-00049) and are currently under review. The permanent DOI will be available upon acceptance of the dataset.

Acknowledgements

This study is supported by the Youth Program of National Natural Science Foundation of China (No. 72301066), Fundamental Research Funds for the Central Universities, China (No.2242024RCB0048), Natural Science Foundation of Jiangsu Province (No. BK20241753), Wuxi Science and Technology Development Fund Project (No. K20231014), SEU Innovation Capability Enhancement Plan for Doctoral Students (No.CXJH_SEU 26206).

Declaration of competing interest

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

Author biography

Chengqi Liu received the B.S. degree in Computer Science from Southeast University in 2023. He is currently pursuing the Ph.D. degree in Transportation Engineering at the Southeast University. His research interests include artificial intelligence and high-performance computing for traffic simulation and control.

Qiang Ma is currently pursuing a master's degree at Southeast University. His main research interests include ROS vehicles and vehicle formation.

Xiwu Wang is currently pursuing a master's degree at Southeast University. His main research interests include car-following models and communication technologies.

Qiang Sun is currently pursuing a master's degree at Southeast University. His main research interests include mesoscopic traffic simulation and traffic flow theories.

Yinke Sun is currently an undergraduate student at the College of Artificial Intelligence, Nankai University, pursuing the B.S. degree in Intelligent Science and Technology. His research interests include computer vision, image processing, and embedded systems.

Zhiyuan Liu received the Ph.D. degree in transportation engineering from the National University of Singapore in 2011. He is currently a Chair Professor at Southeast University, China. His research interests include high-performance computing and optimization for complex urban transportation systems, transportation big data analytics, multimodal foundation models, and transportation network planning and management.

Nan Zheng received the B.Sc. degree in transportation engineering from Southeast University, China, in 2007, the M.Sc. degree in transportation engineering from the Delft University of Technology in 2009, and the Ph.D. degree in transportation engineering from the École Polytechnique Fédérale de Lausanne (EPFL) in 2014. He is currently a Senior Lecturer with Monash University. His research interests include traffic flow theory, traffic big data, urban traffic operation and control, multi-modal, and new-generation traffic management.

Kai Huang earned his Ph.D. degree in transportation engineering from the Department of Civil Engineering, Monash University. He is currently an Associate Professor at Southeast University. His research interests include transportation system optimization, vehicle system simulation and data-driven system analysis.