FACTA UNIVERSITATIS
Series: **Mechanical Engineering**
https://doi.org/10.22190 FUME250508031X

Original scientific paper

SCENARIO-LEVEL HIERARCHICAL ENERGY MANAGEMENT CONTROL STRATEGY FOR HYBRID ELECTRIC VEHICLE

Feng Xiao[1], Ruiqiang Zhang[1], Chunyang Qi[1,2]

[1]College of Automotive Engineering, Jilin University, China
[2]State Key Laboratory of Advanced Design and Manufacturing Technology for Vehicle, Hunan University, China

Abstract. *To address the challenge that single-scenario solutions often fail to adequately handle the complex and dynamic mixed environments that drivers regularly navigate, this study aims to leverage learning-based algorithms to design energy management control strategies tailored to the unique characteristics of different driving scenarios. The goal is to achieve precise matching and efficient execution of the driving strategies. The main research focus of this study is developing a scene-level hierarchical energy management control strategy (SHEMS) framework for hybrid electric vehicles in mixed driving environments. In the car-following scenario, to address the challenges of reward function design and the impact of environment and driver habits, an adaptive strategy learning strategy with imitation learning is proposed. To overcome issues of suboptimal expert knowledge and the curse of dimensionality, optimization factors are added. For the intersection scenario, aiming at the challenge of reward sparsity caused by the scarcity of traffic signals and safety incentives, an additional reward mechanism is innovatively proposed to enrich the reward function. The experimental results demonstrate that the SHEMS significantly reduces fuel consumption. Specifically, the PPO, A2C, A3C, and DQN-based strategies achieved 7.52%, 5.29%, 9.6%, and 5.93% reductions, respectively. Taking the DQN algorithm as an example, the emissions of harmful gases CO, HC, PM and NOx are reduced by 18.3%, 14.23%, 16.94%, and 20.9%, respectively, after layering.*

Key words: *Hybrid electric vehicle, Mixed scenarios, Intersection scenarios, Reinforcement learning*

Received: May 08, 2025 / Accepted September 29, 2025
Corresponding author: Chunyang Qi
National Key Laboratory of Automotive Chassis Integration and Bionics, Jilin University, Changchun 130025, China
E-mail: qichunyang@jlu.edu.cn

1. INTRODUCTION

Among various factors influencing vehicle energy consumption, the driving behavior patterns of drivers have been widely recognized to have a significant impact of up to 30 % on the fuel economy of vehicles [1-3]. Further in-depth research by the National Renewable Energy Laboratory in the United States has further consolidated this finding, revealing that the implementation of reasonable and efficient driving behavior strategies can achieve a significant effect of around a 20 % reduction in fuel consumption [4, 5].When addressing the energy management control strategy problem in the intersection scenario, Yu et al. [6] utilized car-following models and real-time optimal control to optimize the trajectories of connected and automated vehicles (CAVs), thereby minimizing the total fuel consumption of each CAV during the current control period. The fuel economy of hybrid electric vehicles is closely related to the vehicle's speed distribution, which directly affects the required total energy consumption [7]. The latest research surveys have shown that by combining energy management control techniques and speed planning, energy consumption can be effectively reduced, achieving a 15 % energy-saving effect [8]. It not only helps enhance the market attractiveness of products, but also can bring more substantial economic benefits [9]. Hu et al. [10] developed an integrated optimal controller for hybrid electric vehicles equipped with vehicle-infrastructure communication, which achieved up to 16.9% improvement in fuel efficiency on rolling terrain. Liu et al. [11] employed a combination weighting method and K-means clustering to develop an eco-driving evaluation model based on multi-source data. In this case, calculating the reference speed curve to avoid idling before the intersection can ensure relatively good energy-saving performance [12].

In recent years, with the rapid development of artificial intelligence technology, many studies have introduced learning-based methods into the energy management control strategy problem, which have also demonstrated the strong adaptability and nonlinear system optimization capabilities of learning-based methods [13-16]. Yeom et al. [17] cleverly integrated deep reinforcement learning and model predictive control techniques, significantly improving the energy efficiency and economics of electric vehicles. This innovative hybrid approach, particularly the precise optimization of the speed control strategy for electric vehicles through the model predictive control algorithm, not only brings technological innovation to the current electric vehicle field, but also provides valuable theoretical guidance and practical examples for the future planning of autonomous. Addressing the dual objective of reducing fuel consumption during the driving process and simultaneously enhancing driving comfort. He et al. [18] developed a multi-objective intelligent eco-driving strategy for plug-in hybrid electric vehicles based on a multi-head deep Q-network deep reinforcement learning approach, which achieves energy consumption optimization while ensuring autonomous driving safety. Ozatay et al. [19] proposed a simplified reinforcement learning method, which learns the trends and characteristics of the driver's reference speed online and dynamically adjusts the speed limit to increase the driver's likelihood of following the reference speed.

In the frontier field of exploring hierarchical energy management control strategies, Wang et al. [20] took an important step forward by carefully conceiving and designing an online control strategy for hybrid electric vehicles. *SHEMS* aims to comprehensively improve fuel economy, significantly reduce exhaust emissions, and simultaneously ensure a high standard of driving safety. At the low-level execution layer of the strategy, the focus

is on the optimization of power allocation between the internal combustion engine and the battery module, ensuring the rationality and efficiency of energy flow through precise control. At the high-level decision-making layer, the intelligent system is fully responsible for the accurate control and optimization of the vehicle's speed. Guo et al. [21] combined energy management control strategies with reinforcement learning algorithms, proposing a hierarchical control framework-based efficient and coordinated optimization strategy for autonomous hybrid power tracked vehicles, which achieved simultaneous optimization of precise path tracking and energy management. Aiming at safety and comfort, Liu et al. designed an algorithm with a two-layer structure. The upper-level controller uses a grey neural network to predict the future speed trend of the lead vehicle, while the lower-level adopts an adaptive equivalent consumption minimization strategy to implement the hierarchical energy management control strategy [22]. Peng et al. [23] proposed a multi-lane hierarchical optimization algorithm based on a predictive control framework, which plans the optimal speed and lane-changing behavior by considering the vehicle's power demand, driving comfort, and safety at the upper level. At the lower level, dynamic programming is used to design the energy management by tracking the optimal speed. Li et al. [24] studied a data-driven optimal energy management control strategy for plug-in hybrid electric vehicles, which has good performance and computational efficiency in speed planning and energy management. This algorithm also has a hierarchical structure and introduces two data-driven algorithms based on high-fidelity neural networks as the evaluation model and system model for speed optimization, reducing the requirement for complex powertrain system models.

Although there has been extensive research on energy management control and driving style, in the context speed optimization for long-distance urban roads, are relatively limited [25-27]. Chen et al. proposed an assistance system that uses a model predictive control design to create a simplified powertrain model-based advisor system, which not only provides speed profiles but also real-time recommendations for advanced driving modes such as cruising and coasting [28]. Considering the diversity of driver behavior, Ma et al. proposed a novel fuel-efficient driving strategy and investigated its impact on the fuel efficiency of a human-driven vehicle fleet. In mixed traffic, where multiple vehicles share the road, the strategy of the vehicles is realized through vehicle connectivity and longitudinal dynamic control, thereby significantly reducing the vehicle's fuel consumption by avoiding unnecessary braking and acceleration operations [29].

From the current research on energy management control strategies, a significant trend is that a large number of studies have focused on optimization strategies for specific scenarios, such as intersection management and energy management control strategies for highways [30-32]. However, drivers' daily commutes often traverse through complex and ever-changing mixed scenarios, and solutions for a single scenario may prove inadequate in addressing comprehensive road conditions, thus failing to ensure the optimality and adaptability of the strategies. Therefore, in the construction of intelligent transportation systems, the development of an algorithm that can flexibly cope with mixed driving scenarios has become increasingly urgent and important. This paper addresses this challenge by delving into the energy management control strategy for hybrid electric vehicles in mixed scenarios, and innovatively proposing *SHEMS*. The core of this method lies in the customization of energy management control strategies according to the characteristics of different driving scenarios, thereby achieving precise matching and efficient execution of the strategies.

The specific contributions of this paper are as follows:

1. Facing the actual situation of hybrid electric vehicles driving in mixed scenarios, SHED framework for mixed scenarios has been constructed.

2. Regarding the challenge of reward function design, in the car-following scenario, an adaptive policy learning strategy based on imitation learning and a vehicle car-following strategy incorporating driving style are proposed. Considering the challenges of non-optimality of expert knowledge and high-dimensional features, an optimization factor addition strategy is also proposed.

3.To address the sparsity of the reward function in the intersection scenario, a method of adding supplementary rewards to compensate for the sparse reward function and a state prediction calibration method are proposed to solve the problems of sparse reward functions and the uncertainty of the traffic scenario and the learning environment of the intelligent agent.

The remaining parts of the paper are as follows: Section 2 is the problem description, which includes the modeling of hybrid electric vehicles. Section 3 addresses the influence of the traffic environment and driver habits on the driving strategy, and proposes an adaptive policy learning method based on imitation learning and a vehicle car-following strategy incorporating driving style. Section 4 discusses the energy management control strategy in the intersection scenario. Section 5 verifies the algorithm advantages from both simulation and hardware-in-the-loop experiments. Section 6 summarizes the entire paper and looks ahead to the future.

2. PROBLEM DESCRIPTION

2.1 HEV Modeling

The research focuses on a parallel HEV with a P2 configuration, as depicted in Fig.1. The P2 HEV model offered by the Matlab/Simulink 2020b community is utilized and modified in the control component to enhance simulation precision and repeatability. Table 1 contains detailed information about the model parameters.

Table 1 Other parameters of the vehicle

Symbol	Parameter	Values
Engine	Maximum power	92 kW
	Maximum torque	175 Nm
	Maximum speed	6500 rpm
Motor	Maximum power	30 Kw
	Maximum torque	200 Nm
	Maximum speed	6000 rpm
Battery	Capacity	5.3 Ah
	Voltage	266.5 V

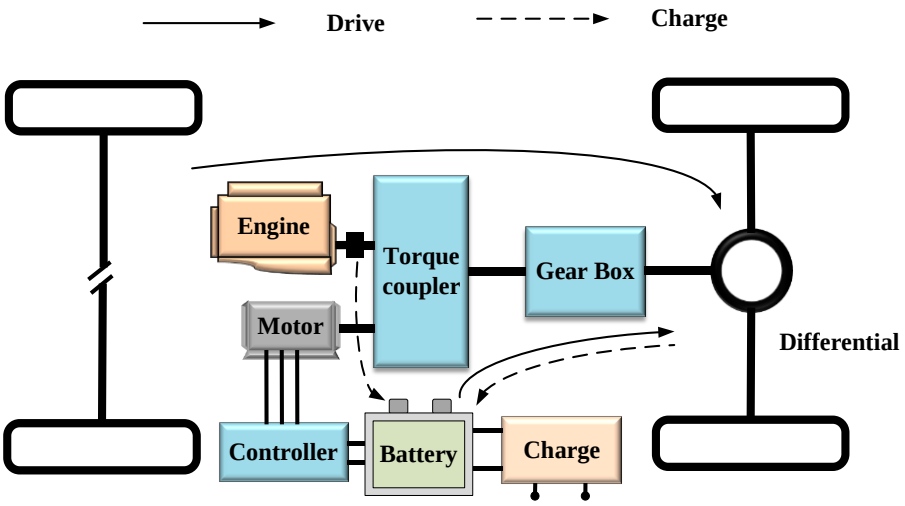


Fig. 1 Vehicle transmission configuration

2.1.1 Vehicle Dynamics Modeling

The longitudinal force balance equation of the vehicle is shown in Eq. (1). The equation is divided into four parts: rolling resistance F_f, aerodynamic drag F_w, slope resistance F_i, and inertial force F_a.

$$\begin{cases} F = F_f + F_w + F_i + F_a \\ F_f = W_f = G \cdot f \\ F_w = \frac{1}{2}\rho \cdot A_f \cdot C_D \cdot v^2 \\ F_i = G \cdot i \\ F_a = \delta \cdot m \cdot a \end{cases} \tag{1}$$

where G represents the vehicle weight, f is the rolling coefficient, ρ indicates the air density, A_f represents the frontal area of the vehicle, C_D donates the drag coefficient, i is the slope angle, δ means the rotational mass conversion factor, and a indicates the longitudinal acceleration of the vehicle.

The power of the hybrid electric vehicle is provided by the engine and the electric motor, which can be calculated as:

$$P_{dem} = (P_{en} + P_{bat} \cdot \eta_m)\eta_T \tag{2}$$

where P_{en} represents the engine power, P_{bat} is the battery power, η_m donates the electric motor efficiency, and η_T indicates the transmission efficiency.

2.1.2 HEV Components Modeling

The engine modeling adopts a static MAP approach. Based on the engine's speed and torque parameters, the corresponding fuel consumption at that speed and torque can be directly obtained from the MAP chart. In this paper, the transient response can be neglected, and the specific formula is:

$$m_{fuel} = \int_0^t \dot{m}_{fuel}\left(T_{engine}, \omega_{engine}\right) dt \tag{3}$$

where m_{fuel} and $\dot{m}_{fuel}$ represent the fuel consumption and the fuel consumption rate, respectively, and ω_{engine} is the engine speed. The harmful gas emissions from the engine are modeled with the RSM model.

The electric motor has two operating modes: motor drive mode and regenerative braking mode, which functions as a traction motor and a generator, respectively. The power output is represented as:

$$P_{motor} = \begin{cases} \frac{T_{motor}\omega_{motor}}{\eta_{motor}}, & T_{motor} \geq 0 \\ T_{motor}\omega_{motor}\eta_{motor}, & T_{motor} < 0 \end{cases} \tag{4}$$

where P_{motor}, ω_{motor}, and η_{motor} indicate the electric motor output power, output speed, and efficiency, respectively.

The rate of change of the state of charge (SOC) is given by:

$$d_{SOC} = \frac{I_{battery}(t)}{Q_{battery}} = -\frac{E_{oc} - \sqrt{E_{oc}^2 - 4R_{battery}P_{battery}}}{2R_{battery}Q_{battery}} \tag{5}$$

where $P_{battery}$ represents the battery charge and $Q_{battery}$ is the total battery capacity.

2.1.3 Driver Modeling

The forward simulation modeling approach is applied, which requires the establishment of a driver model to obtain the acceleration and braking signals. The driver model is constructed based on PID control, by adjusting the PID parameters to calculate the pedal opening based on the target vehicle speed and the actual vehicle speed, as shown in Eq. (6):

$$\begin{cases} \Phi_{req} = K_p \cdot e(t) + K_i \cdot \int e(t)dt + K_d \frac{de(t)}{dt} \\ e(t) = v_{Tar} - v \end{cases} \tag{6}$$

where, e represents the deviation between the target vehicle speed and the actual vehicle speed, and K_p, K_i, and K_d mean the parameters of the PID controller, respectively.

2.2 Strategy Assumptions and Scenario Classification

To make the problem more specific and complete the task better, the following assumptions are made:

(1) The connected and automated vehicle (CAV) can communicate in real-time with the infrastructure on the roadside during driving, and the vehicle can obtain traffic information, location information, and signal light information at intersections through communication technology.

(2) Traffic violations by pedestrians and interference from non-motorized vehicles are not considered.

(3) The state of the yellow light is not considered.

In the smart connected traffic environment, the traffic lights at the intersection have the ability to communicate with the vehicles, and the communication range threshold between the signal light and the vehicle can be represented as R_c. When the distance is less

than R_c, the signal light needs to provide timing information, and the strategy at the intersection is triggered. When the distance is greater than R_c, the strategy at the intersection is not needed, and the vehicle drives under its own car-following behavior. When there is no leading vehicle and no intersection ahead, the vehicle is in a free-driving state. In summary, the scenario classification strategy of this paper is as follows:

$$\begin{cases} Eco-driving\ strategy\ for\ intersection\ scenes: & R < R_c \\ Eco-driving\ strategy\ for\ typical\ scenes: & R \geq R_c \\ Eco-driving\ strategy\ for\ free-driving\ scenes: & \begin{matrix} no\ leading\ vehicle\ or \\ no\ intersection\ ahead \end{matrix} \end{cases} \tag{7}$$

2.3 SHED Modeling under Reinforcement Learning

In the free-driving scenario, the following vehicle's speed should not exceed the maximum speed limit of the lane the leading vehicle is in, nor should it exceed the maximum constraint speed of the vehicle. Once the following vehicle accelerates to the maximum permitted speed, it should enter a stable speed control mode and maintain that speed until a different traffic situation is encountered.

The speed limit is given by:

$$F_{Free} = v_{lim}(k) - v(k) \tag{8}$$

$$v_{lim}(k) = min(v_{lane}, v_{max}) \tag{9}$$

The value of $v_{lim}(k)$ is selected as the smaller of the two speeds, which are the lane speed limit v_{lane} and the maximum allowed speed of the following vehicle v_{max}, over the time step k.

The reward function for the strategy in the free-driving scenario is as follows:

$$R_{freedom} = \alpha_1 Fuel + \alpha_2 Security + \alpha_3 Emission \tag{10}$$

If F_{Free} is negative, the reinforcement learning reward function is directly set to -1 to receive a negative reward. Security represents the condition where there is no collision with other traffic environments.

The reinforcement learning state vector is:

$$S_{t\_freedom} = \left[V_{ego}, D_{ego}, i, SOC, D_{V2S}\right]^T \tag{11}$$

where V_{ego} indicates the reference vehicle speed, D_{ego} represents the distance to the leading vehicle, i means the road slope, SOC donates the state of charge of the battery, and D_{v2s} is the distance to the destination.

Under the typical scenario condition, the car-following model algorithm can integrate environmental characteristics and driver behavior characteristics while ensuring safety. By incorporating energy management and pollutant emissions, the following reward function is obtained:

$$R_{following} = R_{freedom} + F_{following} \tag{12}$$

The reinforcement learning state vector is:

$$S_{t\_following} = \left[V_{ego}, D_{ego}, V_{pre}, a_{pre}, D_h, i, SOC, D_{V2S}\right]^T \tag{13}$$

where, V_{pre} represents the speed of the leading vehicle, a_{pre} indicates the acceleration of the leading vehicle, D_h means the distance to the leading vehicle.

In the intersection scenario, the optimization function is composed of objectives for safety, energy management, and harmful substance emission reduction. The reinforcement learning reward function is as follows:

$$R_{s_{following}} = \beta_1 Fuel + \beta_2 Security + \beta_3 Emission + \beta_4 Trafficlight + F(s_t, a_t) \quad (14)$$

The reinforcement learning state vector is:

$$S_t = \begin{bmatrix} V_{ego}, D_{ego}, V_{pre}, a_{pre}, D_h, i, SOC, D_{V2S}, \\ t_{r\_rem}, t_{g\_rem}, V_{light\_min}, V_{light\_max} \end{bmatrix}^T \quad (15)$$

That is the hierarchical scenario structure for the SHED algorithm.

The flow chart of the algorithm is shown in Fig. 2. The energy management control strategies can be divided into three typical scenarios: free-driving, car-following, and intersection. In the free-driving scenario, the factors affecting consist of fuel consumption, driving safety, and emissions. The reward function only considers these three factors. For the typical car-following scenario, the strategy is built upon the free-driving case, with additional factors such as the adaptive factor, mutual information, high-dimensional features, and driver's style. The corresponding reinforcement learning state vector is expanded accordingly. Under the intersection scenario, the strategy extends the free-driving case by incorporating an additional reward that combines self-supervised internal reward functions, calibration formulas, and traffic signal status. The reinforcement learning state vector is further expanded to capture these intersection-specific elements.

3. Adaptive Vehicle Following Strategy Based on Imitation Learning

The adaptive strategy learning approach that combines imitation learning, as well as an imitation learning strategy that incorporates the driver's style are proposed. The factors influencing vehicle car-following can be broadly categorized into two aspects: The impact of the environment itself on the driver's behavior and the inherent influence of the individual driver's behavioral habits on the car-following strategy.

3.1 Environmental Factors Influencing Driver Behavior: Adaptive Strategy Learning

For complex and ambiguous environments, imitation learning strategies, which overcome the obstacle of manually specifying an appropriate reward function, have the potential to outperform reinforcement learning strategies [33-35]. In real-world tasks, drivers may exhibit multi-intent behavior when following a lead vehicle, and different drivers have varying driving habits [36,37]. These diverse driving habits make it challenging to select a suitable reward function. The car-following model with adaptive

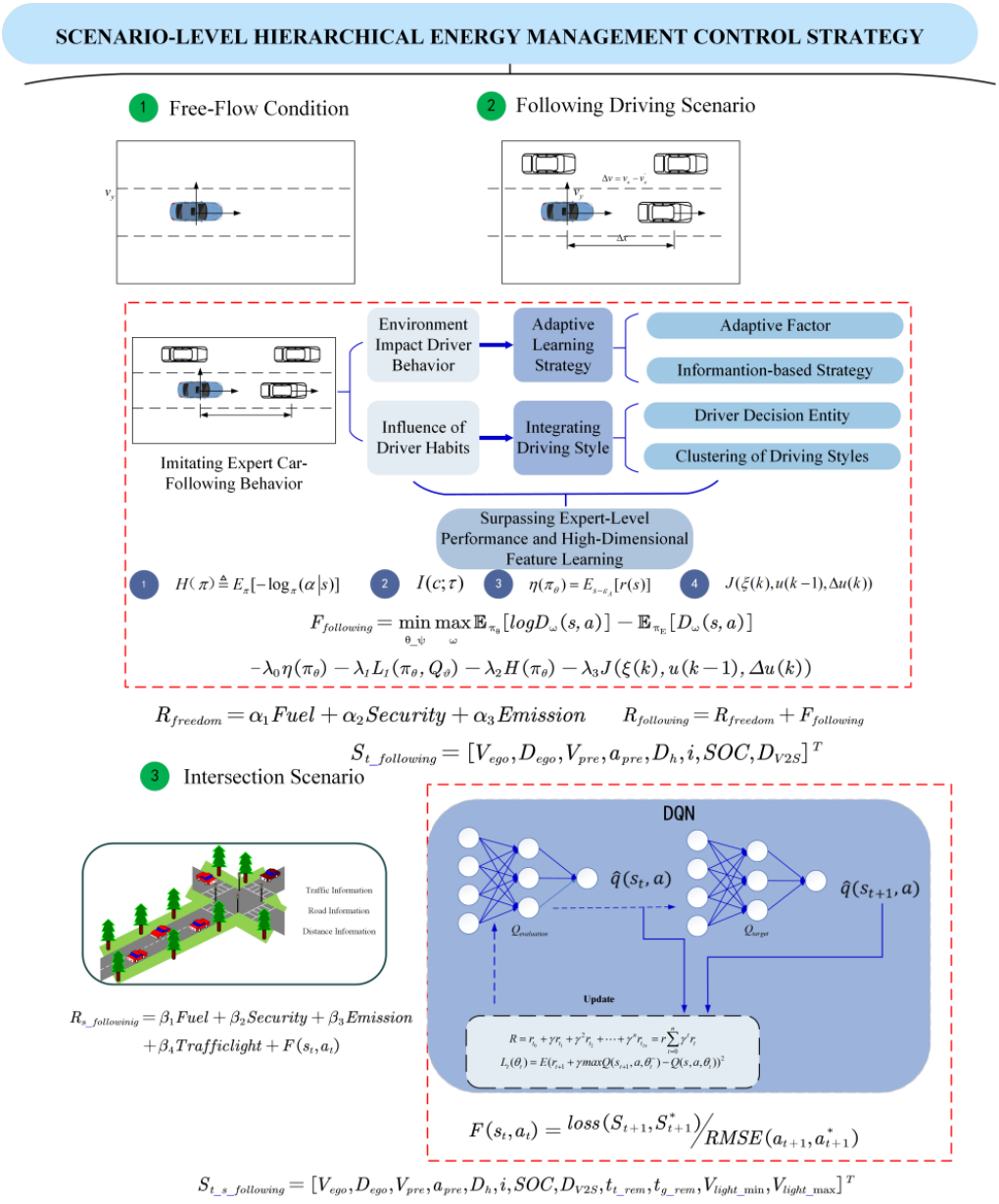


Fig. 2 Scene-level hierarchical energy management control strategy

factor that addresses the inherent phenomenon of environmental influences on driver behavior is introduced in this section. The main idea is to construct an adaptive GAN framework based on the principles of GAN, which trains the car-following policy directly without estimating the reward function. As is shown in Eq. (16):

$$\min_{\pi} \max_{D\in(0,1)^{S\times A}} \mathbb{E}_{\pi}[log\, D(s,a)] + \mathbb{E}_{\pi E}\big[log\big(1 - D(s,a)\big)\big] - \lambda H(\pi) \tag{16}$$

where, π_E represents the expert policy, π indicates the policy that needs to be learned, D serves as the discriminator, and $H(\pi)$ means the causal entropy, which is shown in Eq. (17):

$$H(\pi) \triangleq E_{\pi}[-log\, \pi(\alpha \mid s)] \tag{17}$$

Despite the adaptive policy is model-free, it still requires interacting with the environment during training. Due to the presence of stochastic factors, the resulting trajectories can exhibit significant variations across different individuals. Even the same individual may make different decisions when facing the same situation, leading to the generation of multiple distinct policies.

In this scenario, a set of expert policies: $\pi_E = \{\pi_E^0, \pi_E^1, \ldots\}$ and the process of generating expert trajectories: $s_0 \sim \rho_0 c \sim p(c),\ \pi \sim p(\pi \mid c)_0 a_t \sim \pi(a_t \mid s_t),\ s_{t+1} \sim P(s_{t+1} \mid a_t, s_t)$ are defined, where the adaptive variable c with a prior probability distribution $p(c)$ are also introduced. The goal of the algorithm is to recover the policy $\pi(a \mid s, c)$, under the adaptive variable c. To ensure a tighter coupling between the adaptive factor and the policy, and to increase their correlation, we incorporate mutual information into the optimization function.

$$\begin{aligned} I(c;\tau) &= H(c) - H(c \mid \tau) \\ &= H(c) + \int_c \int_\tau p(c,\tau)\, log\, p(c,\tau)\, dcd\tau \\ &= H(c) + \int_c \int_\tau p(c)(\tau \mid c)\, log\, p(c \mid \tau)\, dcd\tau \\ &= H(c) + \mathbb{E}_{c\sim p(c), a\sim\pi(\cdot\mid S, c)}[log\, p(c\mid\tau)]\big] \end{aligned} \tag{18}$$

Incorporating the adaptive factor into the optimization objective function, the equation can be formulated as:

$$\min_{\theta\_\psi}\max_{\omega} \mathbb{E}_{\pi_\theta}[log\, D_\omega(s,a)] + \mathbb{E}_{\pi_E}\big[log\big(1 - D_\omega(s,a)\big)\big] - \lambda_1 I(c;\tau) - \lambda_2 H(\pi_\theta) \tag{19}$$

Given the difficulty of computing the posterior probability $p(c\mid \tau)$ during the optimization process, the Q-value is applied directly as a replacement, as shown in the following equation:

$$L_1(\pi, Q) \le I(c;\tau) \tag{20}$$

$$\min_{\theta\_\psi}\max_{\omega} \mathbb{E}_{\pi_\theta}[log\, D_\omega(s,a)] + \mathbb{E}_{\pi_E}\big[log\big(1 - D_\omega(s,a)\big)\big] - \lambda_1 L_1(\pi, Q) - \lambda_2 H(\pi_\theta) \tag{21}$$

3.2 Surpassing Expert-Level Performance and Learning High-Dimensional Features

In the car-following task, the limitations of the expert trajectories input to the network can lead to the situation where the error of the learned policy exceeds that of the expert policy. For the car-following task, it is difficult to define a suitable reward function. From the perspective of the policy, the problem becomes relatively more straightforward. The car-following task model from the policy standpoint is shown in Eq. (22):

$$\eta(\pi_\theta) = \mathbb{E}_{s\sim\varepsilon_A}[r(s)] \tag{22}$$

The objective optimization function can be transformed as follows:

$$\begin{aligned}\min_{\theta\_\psi}\max_{\omega}\mathbb{E}_{\pi_\theta}[log\ D_\omega(s,a)] + \mathbb{E}_{\pi_E}\big[log\big(1-D_\omega(s,a)\big)\big] \\ -\lambda_\theta\eta(\pi_\theta) - \lambda_1 L_1\big(\pi_\theta, Q_\psi\big) - \lambda_2 H(\pi_\theta)\end{aligned} \tag{23}$$

where λ represents a hyper-parameter. The reinforcement function for policy optimization consists of two components: the surrogate reward and the discriminator, which aims to mimic the expert. This approach allows to overcome the limitation of imitation learning.

To effectively learn strategies for high-dimensional inputs, the original GAIL framework has been improved as shown in Eq. (24). To address traditional GAN networks suffer from issues such as gradient vanishing and mode collapse, the objective function has been further enhanced by incorporating Wasserstein GAN techniques, as follows:

$$\begin{aligned}R_F = \&\min_{\theta\_\psi}\max_{\omega}\mathbb{E}_{\pi_\theta}[D_\omega(s,a)] - \mathbb{E}_{\pi_E}[D_\omega(s,a)] \\ -\lambda_\theta\eta(\pi_\theta) - \lambda_1 L_1\big(\pi_\theta, Q_\psi\big) - \lambda_2 H(\pi_\theta)\end{aligned} \tag{24}$$

where the discriminator network D_ω and the posterior approximation network are treated as separate networks.

3.3 Optimizing Car-Following Model by Integrating Driver's Driving Style

During vehicle operation, it is important to consider not only the impact of the driving environment on the model, but also the need to satisfy the driver's comfort and safety in the car-following task. In this section, the essential characteristics of driving behavior is explored under different driving modes, taking into account the vehicle's own dynamics. The optimization strategy also incorporates the driver's stylistic characteristics in the car-following task. The driver is the core of the vehicle-road-traffic system. Although the car-following behavior describes the relationship between the lead and the following vehicles, its essence is a description of the driver's behavior in a given traffic environment.

3.3.1 Driver Style Clustering Analysis

To establish a personalized car-following model, the driving data that can reflect the driver's style should be classified based on different driving styles. Combining the inter-vehicle distance and the velocity of the lead vehicle, the Time-To-Collision Inverse (TTCI) is defined as follows:

$$TTCI = \frac{\Delta v}{d} \tag{25}$$

where, Δv represents the difference in velocity between the following vehicle and the lead vehicle, and d donates the distance between the vehicles. In summary, the behavior features consist of the time-to-collision (TTC), time headway, and the absolute values of acceleration and deceleration indirectly reflect the driver's style. The driver's style can be broadly categorized into two main groups: conservative and aggressive. The K-means clustering strategy [38] can be used to cluster the driving data into these two driving style categories.

3.3.2 Integrated Car-Following Strategy Considering Driver Style Features

Given that individuals have different perceptions of comfort, studying the car-following model alone is clearly insufficient. In this section, integrating the driver's style into the car-following strategy is introduced, which enables the car-following model to have personalized characteristics, improving the overall car-following task. This strategy can effectively solve the multi-objective optimization problem in the car-following task. Drivers with the same style tend to maintain similar time head-ways in typical car-following scenarios, but drivers with different styles can have significant differences in time headway. The formula for the inter-vehicle distance can be expressed as:

$$\Delta d_{des} = v_f t_h + d_0 \tag{26}$$

where v_f is the traveling speed of the following vehicle, t_h represents the progression of time, and d_0 means the safe distance when the vehicle is traveling at very low speeds.

The goal of vehicle car-following is for the driver to adjust the vehicle's speed to match the speed of the lead vehicle, and maintain the inter-vehicle distance around the desired value. Inter-vehicle Distance Error $\Delta d_{err}(k)$ is defined as:

$$\Delta d_{err}(k) = \Delta d - \Delta d_{des} \tag{27}$$

where $\Delta d_{err}(k) \to 0, \Delta v(k) \to 0$.

To provide comfort to passengers during car-following, the absolute values of both acceleration and jerk should also be minimized as much as possible. j_k, which represents the rate of change of acceleration is defined as:

$$j(k) = \frac{a(k)-a(k-1)}{\Delta t} \tag{28}$$

where $|a(k)| \to 0, |j(k)| \to 0$.

When solving the optimization problem, certain constraints must be satisfied. To avoid colliding with the lead vehicle, the inter-vehicle distance must satisfy the minimum distance constraint. Ensuring the following vehicle remains in the car-following scenario, this distance should be less than the maximum car-following distance. The minimum and maximum values of velocity, acceleration, and jerk must also be constrained. Model predictive control can handle multi-variable and constrained problems, making it suitable for incorporating the car-following model within an MPC framework. Based on the principles of MPC, the cost function can be defined as:

$$\begin{aligned} J\big(\xi(k), u(k-1), \Delta u(k)\big) = \&\sum_{i=1}^{N_P} \|\eta(k+i \mid k) - \eta_{ref}(k+i \mid k)\|_Q^2 \\ + \sum_{i=1}^{N_c-1} \| \Delta u(k+i \mid k) \|_R^2 \end{aligned} \tag{29}$$

where $\boldsymbol{\eta_{ref}}(k+i \mid k)$ represents the reference vector, and $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the weighted matrices for the control and target variables, respectively, where $\boldsymbol{R}$ is a one-dimensional vector with a value of 10, and $\boldsymbol{Q}$ is a fourth-order square matrix, respectively, which, in the car-following task, can be defined as follows:

$$\begin{aligned} \eta_{ref}(k+i \mid k) &= [\Delta d_{des}(k+i \mid k) \quad 0 \quad 0 \quad 0]^T \\ \Delta u(k+i \mid k) &= j(k+i \mid k)\Delta t \end{aligned} \tag{30}$$

The optimization objective and the constraints for the model predictive control are defined as:

$$s.t.\begin{cases} d_0 \le \Delta d(k) \le d_{max}, \\ v_{min} \le v(k) \le v_{max} \\ a_{min} \le a(k) \le a_{max} \\ j_{min} \le j(k) \le j_{max} \end{cases} \tag{31}$$

Considering driving styles, the aggressive strategy should have safety and comfort, while the conservative driving strategy should have higher safety and comfort. The final cost function is defined as:

$$\begin{aligned} F_{following} = \min_{\theta\_\psi} \max_{\omega} \mathbb{E}_{\pi_\theta}[log\, D_\omega(s,a)] - \mathbb{E}_{\pi_E}[D_\omega(s,a)] - \lambda_0 \eta(\pi_\theta) \\ -\lambda_I L_I(\pi_\theta, Q_\vartheta) - \lambda_2 H(\pi_\theta) - \lambda_3 J\big(\xi(k), u(k-1), \Delta u(k)\big) \end{aligned} \tag{32}$$

4. Reinforcement Learning Optimization of Energy Management Control Strategies in Intersection Scenarios

The factors affecting the driving control strategy at intersections can be divided into two main aspects. The first is the impact of sparse reward functions on the driving control strategy, and the second is the impact of the complexity and uncertainty of intersections on the driving control strategy. To address these issues, two approaches: Self-supervised additional reward function and calibration equation for adjusting the training direction are proposed.

4.1 Reconstruction of the Safety Reward Function

During the actual training process, the safety reward function often suffers from sparsity, which is detrimental to the learning of the intelligent agent. Given the above issues, it is crucial to redesign the safety reward function considering the potential impact of the current behavior on future states. In the context of intelligent and connected vehicles, the vehicle itself can receive information about the phase and distance of the next traffic light. Drivers generally expect the next traffic light to be green, allowing them to quickly pass through the intersection. Based on the remaining time of the red light and the duration of the green light, the speed boundaries at the intersection can be determined as:

$$V_{light\_max} = min\left\{ \begin{matrix} V_{limit}, \\ D_{V2S} \div \left[t_{r\_rem} + k\left(t_{g\_cycle} + t_{r\_cycle}\right)\right] \end{matrix} \right\} \tag{33}$$

$$V_{light\_min} = D_{V2S} \div \left[t_{r\_rem} + t_{g\_cycle} + k\left(t_{g\_cycle} + t_{r\_cycle}\right)\right] \tag{34}$$

where, V_{limit} represents the speed limit of the road. D_{V2S} indicates the distance to the stop line at the intersection. t_{r_cycle} shows the remaining red light time. t_{g_cycle} means the duration of the green light. t_{r_rem} is the duration of the red light. $k = 0,1,2 \ldots, \infty$ represents the delay period. When the vehicle can pass through the next red-green light during the green light time TW, k = 0. If the lower bound of the green light speed boundary exceeds the road speed limit, it means the vehicle cannot pass through the next red-green light. In this case, we can adjust the delay period k to select a suitable green light.

Similarly, when the traffic light is green, the speed boundary can be derived as:

$$V_{light\_max} = \begin{cases} V_{limit}, k = 0 \\ min\left\{ \begin{matrix} V_{limit}, \\ D_{V2S} \div \left[t_{g\_rem} + (k-1)\left(t_{g\_cycle} + t_{r\_cycle}\right) + t_{r\_cycle} \right] \end{matrix} \right\} \end{cases}$$

$$V_{light\_max} \begin{cases} k = 0 \\ k = 0,1,2 \ldots, \infty \end{cases} \tag{35}$$

$$V_{light\_min} = D_{V2S} \div \left[t_{g\_rem} + k\left(t_{g\_cycle} + t_{r\_cycle}\right) \right]$$

where, t_{g_rem} is the remaining green light time.

If the ego vehicle's speed exceeds the speed boundaries, there is a risk of running the red light. The traffic light reward function can be represented as a piece-wise function:

$$r_{light} = \begin{cases} \beta_1 + ln\left(v(n) - V_{light\_max}\right)^2, \text{if } V_{light\_max} < v(n) \text{ and } \beta_1 < \left(v(n) - V_{light\_max}\right)^2 \\ \left(v(n) - V_{light\_max}\right)^2, \quad \text{if } V_{light\_max} < v(n) \text{ and } \left(v(n) - V_{light\_max}\right)^2 \le \beta_1 \\ 0, \quad \text{if } V_{light\_min} \le v(n) \le V_{light\_max} \\ \left(v(n) - V_{light\_min}\right)^2, \quad \text{if } v(n) < V_{light\_min} \text{ and } \left(v(n) - V_{light\_min}\right)^2 \le \beta_2 \\ \beta_2 + ln\left(v(n) - V_{light\_min}\right)^2, \text{if } v(n) < V_{light\_min} \text{ and } \beta_2 < \left(v(n) - V_{light\_min}\right)^2 \end{cases} \tag{36}$$

where, β_1 and β_2 represent the boundary points of the green light speed boundary, respectively. When the vehicle speed exceeds the green light speed boundary, we can calculate a negative reward based on the proposed traffic light reward function, rather than only calculating a sparse reward when the vehicle runs the red light.

4.2 Design of Additional Reward Functions

The reward function is the guiding direction for the entire task, determining the objectives the agent needs to achieve. The reward function can be simply designed as a combination of fuel consumption, safety, and pollution emissions. Additional reward functions to train the neural network are designed innovatively, guiding the vehicle's actions and enabling the hybrid-electric vehicle agent to have the capability of self-exploration, while preventing actions that the agent may regret later. The general approach to improving the reward function is:

$$r'(s_t, a_t) = r(s_t, a_t) + F(s_t, a_t) \tag{37}$$

where, $F(s_t,a_t)$ represents additional reward. $r(s_t, a_t) = \beta_1 r_{Fuel} + \beta_2 r_{Security} + \beta_3 r_{Emission}$ is basic reward function. The recursive equation for the expected cumulative reward of the revised reward function can be expressed as:

$$Q^z_{new}(s_t, a_t) = r^{'}(s_t, a_t) + \gamma E_{r_t, s_t, 1} - E[Q^z_{new}(s_{t+1}, a_{t+1})] \tag{38}$$

According to Eq. (38), after modifying the reward function, the new Q-value of the Markov Decision Process can be derived as:

$$\begin{aligned} Q^{\pi}_{new}(s_t, a_t) &= \mathbb{E}_{r_{i\ge t}, s_{i>t} \sim E}\left[\sum_{i=t}^{T} \gamma^{i-t} r_i^{'}(s_i, a_i)\right] \\ &= \mathbb{E}_{r_{i\ge t}, s_{i>t} \sim E}\left[\sum_{i=t}^{T} \gamma^{i-t} \left(r(s_t, a_t) + F(s_t, a_t)\right)\right] \\ &= Q^{\pi}(s_t, a_t) + \mathbb{E}_{r_{i\ge t}, s_{i>t} \sim E}\left[\sum_{i=t}^{T} \gamma^{i-t} F(s_t, a_t)\right] \end{aligned} \tag{39}$$

As discussed, a self-supervised strategy to compute the additional rewards is designed, allowing the agent to self-explore and complete the ecological tasks at intersections. Self-supervised learning can be viewed as a subset of unsupervised learning, where the agent learns from the inherent relationships in the data without the need for extensive labeled datasets. In the *SHEMS*, the rewards need to be constructed through the automatic generation of pseudo-labels. Specifically, the agent predicts the next state based on the current state and action, and combines the predicted state with the current state to construct the $Loss$ function. The addition of the self-supervised module gives the model a certain level of generalization capability. The self-supervised internal reward functions are shown in Eq. (40) and Eq. (41), where the Model refers to a generic deep learning prediction model:

$$S_{t+1}^{*} = Model(s_t, a_t) \tag{40}$$

$$F(s_t, a_t) = loss(S_{t+1}, S_{t+1}^{*}) \tag{41}$$

The essence of reinforcement learning lies in the ability to predict the next action based on the current state. By executing the action for the next time step, the agent obtains the state for the following time step. The uncertainty in the environment will directly impact the agent's choice of actions, and the agent's actions will also directly influence the state vector. The key to maintaining a stable state is the accuracy of the action values. The accuracy of the action values is regarded as a factor in the calibration formula:

$$r_{calibration} = RMSE(a_{t+1}, a_{t+1}^{*}) \tag{42}$$

where a_{t+1}^{*} ,which represents the action for the next time step, can be calculated by the formula $a_{t+1}^{*} = \text{model2}(a_t, s_t)$. A represents the true action values. By integrating the reinforcement learning calibration and the additional rewards generated through self-supervision, a new additional reward function is updated as Eq. (43):

$$F(s_t, a_t) = {loss(S_{t+1}, S_{t+1}^{*})}\Big/{RMSE(a_{t+1}, a_{t+1}^{*})} \tag{43}$$

The final reward function for the self-supervised reinforcement learning calibration method is shown in Eq. (44):

$$R_{s_{following}} = \beta_1 Fuel + \beta_2 Security + \beta_3 Emission + \beta_4 Trafficlight + F(s_t, a_t) \tag{44}$$

5. Experimental Results

5.1 Results of the Car-Following Experiment

5.1.1 Data Process

In this study, the car-following model training utilized the high-resolution Waymo open dataset [39]. Given the measurement errors of the equipment, which may have a significant impact on the analysis of trajectory data, it is necessary to establish an appropriate noise filtering mechanism in the numerical calculation process. We opted to apply the widely-

used Savitzky-Golay filter [40], which calculates the smoothed data by employing polynomial fitting and interpolation methods to address this issue. As expresses in Eq. (45):

$$g_i = \sum_{n=-n_L}^{n_R} c_n f_{i+n} \tag{45}$$

where, f_{i+n} represents the smoothed signal, n_L denotes the number of data points to the left of the current data point, n_R indicates the number of data points to the right of the current data point, and c_n means the weight values.

The specific data processing approach involves using a second-order Savitzky-Golay filter to filter each speed data point. The acceleration and deceleration are calculated based on the filtered speed data. Further, an additional smoothing method is employed to eliminate any residual noise.

5.1.2 Results of the Simulation Experiments

To verify the accuracy and reasonableness of the car-following results, an analysis of the simulation outcomes is conducted. 80 % of the Waymo data is selected as the training set to train the model, and the remaining 20 % of the data is designed for testing. The training parameters are shown in Table 2.

Table 2 Car-following model training parameters

Parameters	Value
Learning rate	0.001
Discount Factor	0.99
Training Episodes	400

Analysis of the time-speed and time-distance curves presented in Fig. 3, depicting three distinct operating scenarios, reveals that the proposed algorithm exhibits highly satisfactory car-following behavior. Critically, the algorithm consistently maintains the following vehicle within a safe distance envelope relative to the leader. Furthermore, it demonstrates a dynamic and responsive capability, as the following vehicle's speed continuously adapts to fluctuations in the leading vehicle's speed. This observed performance effectively confirms the model's operational validity. To quantitatively assess and clearly illustrate the differences in speed tracking accuracy between the following vehicle and the leading vehicle across the evaluated models, the Sum of Squared Errors (SSE) was calculated for each dataset. The comparative results highlight significant performance variations: the average SSE values for the Optimal Speed Model, the IDM (Intelligent Driver Model), the Stimulus-Response Model, and the Safe Distance Model are 24.2, 15.08, 7.52, and 3.1, respectively. Crucially, the proposed model achieves a substantially lower SSE of 2.89. This demonstrably smaller error value underscores the superior effectiveness of the novel approach in accurately replicating the nuanced car-following strategies characteristic of expert human driving.

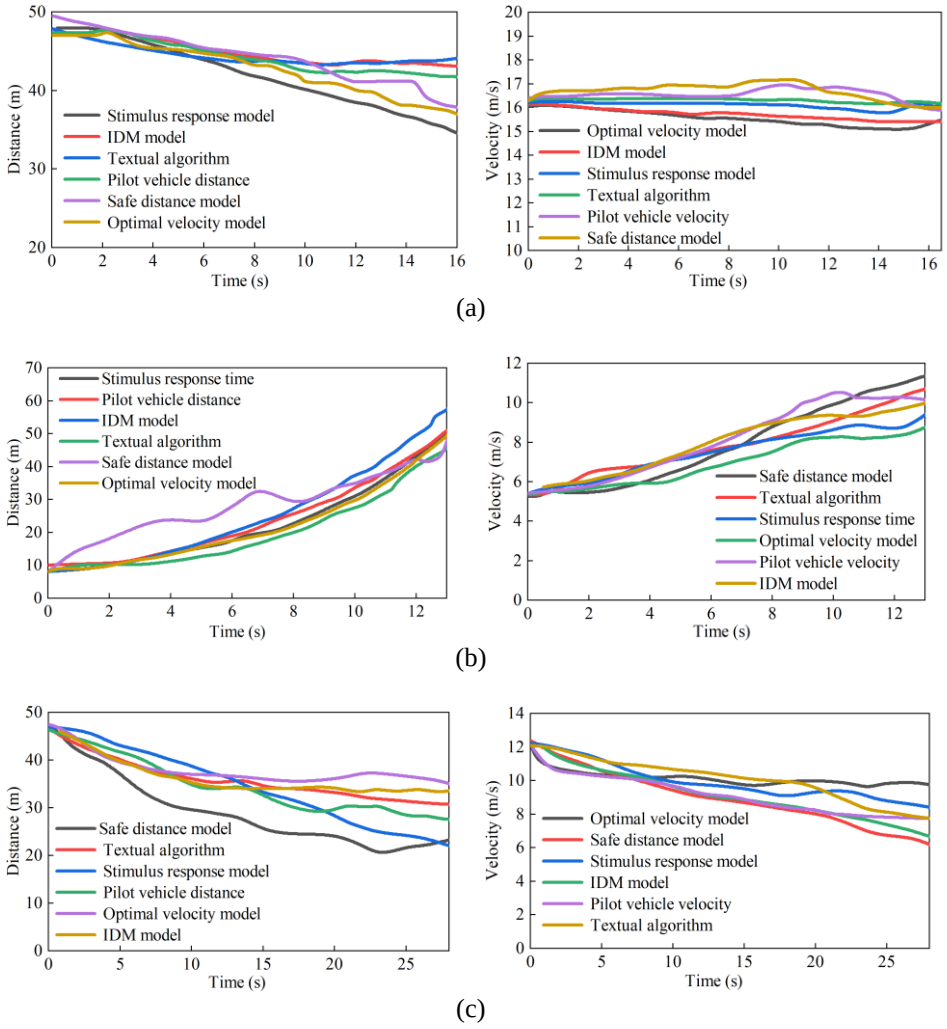


Fig. 3 Time-distance and Time-velocity plots for car-following behavior: (a) Steady-state condition; (b) Acceleration condition; (c) Deceleration condition

To further evaluate the conformity of the car-following model with the expert algorithm, the time-speed and time-distance curves for the algorithm proposed in this chapter, the expert algorithm, and the leading vehicle are presented in Fig. 4. The proposed algorithm is closer to the leading vehicle in certain regions compared to the expert algorithm. In some road segments, the proposed algorithm even surpasses the expert algorithm, and the consistency between the vehicle speed and the leading vehicle is also more closely aligned than the expert algorithm.

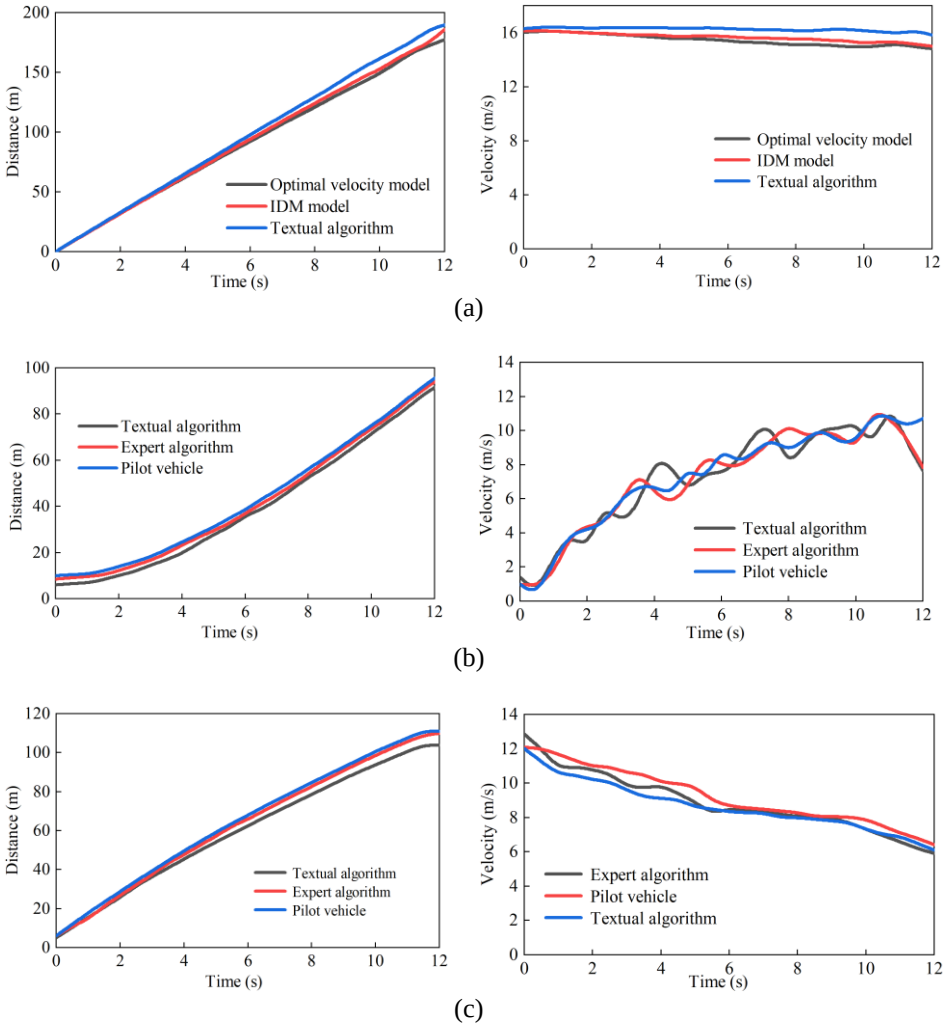


Fig. 4 Comparison of the algorithm in this paper to an Expert-Level algorithm: (a) Steady-state condition; (b) Acceleration condition; (c) Deceleration condition

In order to quantify the results shown in Fig. 4, the speed difference between the vehicle model trained with the two different algorithms and the leading vehicle are calculated, as shown in Table 3. To more clearly illustrate the performance gap between the proposed algorithm and the expert algorithm, the average values of the speed differences are calculated for each case with 0.38, 0.0026, 0.21. The results indicate that the algorithm developed in this chapter can sometimes outperform the expert algorithm.

Table 3 Quantitative comparison of speed metrics between the algorithm in this paper and an expert-level

	Steady-state condition (m/s)			Acceleration condition (m/s)			Deceleration condition (m/s)		
Time (s)	Expert	Paper	Difference	Expert	Paper	Difference	Expert	Paper	Difference
0	0.099	0.148	0.049	0.000	0.050	0.050	0.752	-0.071	-0.823
0.5	0.049	0.248	0.200	0.573	0.112	-0.461	-0.032	-0.780	-0.748
1	-0.001	0.298	0.299	-0.773	0.120	0.893	-0.832	-1.135	-0.303
1.5	-0.001	0.347	0.348	-0.021	0.110	0.131	-0.355	-0.851	-0.496
2	-0.050	0.396	0.446	0.204	-1.210	-1.414	-0.177	-0.780	-0.603
2.5	0.049	0.446	0.397	-0.001	1.210	1.211	-0.390	-0.851	-0.461
3	0.000	0.544	0.544	-0.240	-1.410	-1.170	-0.774	-0.993	-0.218
3.5	0.100	0.541	0.442	0.891	-1.220	-2.111	-0.832	-1.348	-0.516
4	0.199	0.538	0.339	-0.376	1.630	2.006	-0.142	-0.922	-0.780
4.5	0.150	0.637	0.487	-0.509	1.800	2.309	-0.567	-0.922	-0.355
5	0.200	0.537	0.338	-1.233	-1.500	-0.267	-0.893	-1.206	-0.313
5.5	0.299	0.587	0.288	1.598	0.430	-1.168	-0.761	-0.496	0.265
6	0.300	0.588	0.288	-1.054	-1.100	-0.046	-0.176	-0.355	-0.179
6.5	0.349	0.639	0.290	-0.177	0.200	0.377	-0.113	-0.271	-0.158
7	0.300	0.590	0.290	-0.200	1.200	1.400	-0.170	-0.229	-0.059
7.5	0.398	0.640	0.242	-0.101	0.800	0.901	-0.170	-0.350	-0.180
8	0.399	0.640	0.242	1.679	-1.000	-2.679	-0.228	-0.294	-0.066
8.5	0.399	0.640	0.241	0.075	-0.200	-0.275	-0.026	-0.166	-0.140
9	0.351	0.838	0.487	-0.211	0.120	0.331	-0.108	-0.265	-0.158
9.5	0.450	0.839	0.389	0.514	0.760	0.246	-0.315	-0.315	0.000
10	0.297	0.941	0.644	0.644	1.23	0.586	-0.600	-0.600	0.000
10.5	0.309	0.730	0.421	0.421	-2.1	-2.521	-0.550	-0.500	0.050
11	0.136	0.660	0.524	0.524	0.79	0.266	-0.500	-0.200	0.300
11.5	0.192	0.966	0.774	0.774	-0.82	-1.594	-0.600	-0.300	0.300
12	0.200	0.814	0.614	0.614	-3.06	-3.674	-0.470	-0.300	0.170

5.2 Simulation Results in a Mixed Scenario

The simulation is conducted on a PC with an i5-12400 CPU and 16 GB of memory, running in MATLAB and SUMO environments. The following vehicle is modeled as a plug-in hybrid electric vehicle (PHEV) with intelligent connected vehicle (ICV) capabilities. The initial conditions for the vehicle are a battery SOC of 0.65 and a full fuel tank. The simulation results with the proposed car-following algorithm are presented in Fig. 5. The blue solid line represents the speed of the leading vehicle, while the red dashed line depicts the speed of the following vehicle after the optimization of the car-following algorithm. The speed-tracking results demonstrate that the proposed algorithm is able to closely follow the leading vehicle's velocity profile.

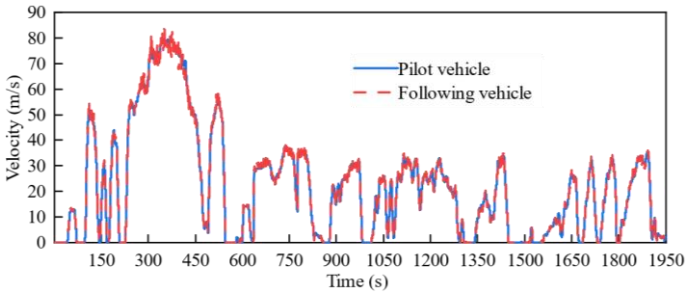
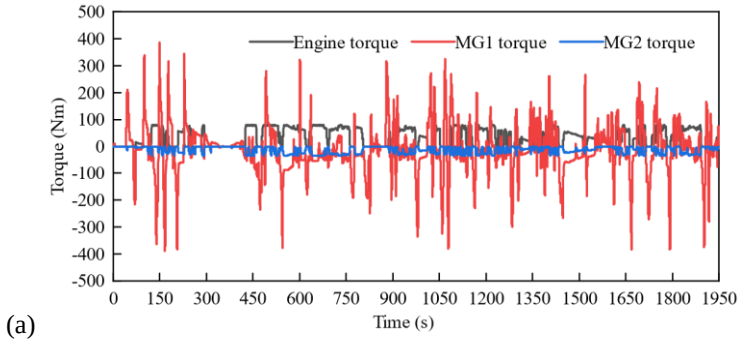
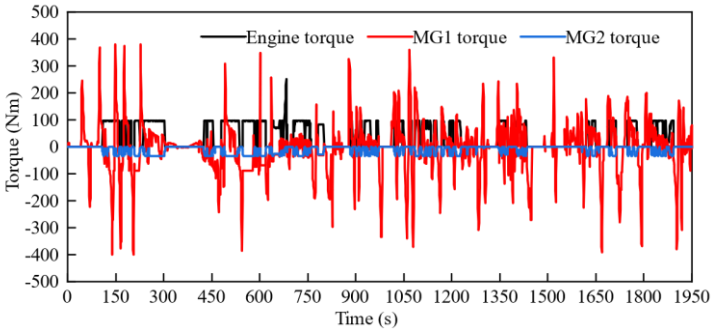


Fig. 5 Speeds of the lead vehicle and the following vehicle

After the implementation of the global car-following algorithm, the strategy at intersections is further integrated, resulting in the engine torque distribution under the given driving conditions, as shown in Fig. 6. The torque distribution of the engine, generator, and electric motor in the non-hierarchical case are presented in Fig. 6(a). In contrast, the torque distribution of the engine, generator MG1, and electric motor MG2 under the hierarchical control strategy is illustrated in Fig. 6(b).



(a)



(b)

Fig. 6 Engine torque, MG1 torque, and MG2 torque: (a) Before layering; (b) After layering

The SOC trajectory under the given driving conditions is illustrated in Fig. 7. The overall trend shows a decrease in SOC over time. In the non-hierarchical case, the battery

SOC exhibits a significant drop around the 1100-second mark. In contrast, the hierarchical control strategy is able to maintain a more stable SOC profile throughout the driving cycle.

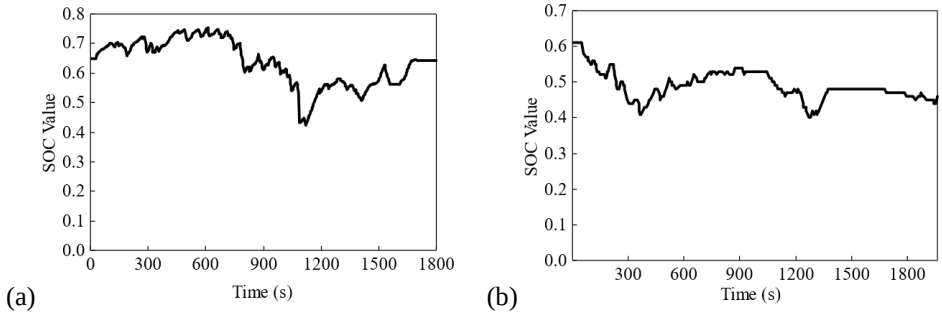


Fig. 7 Comparison of battery SOC before and after hierarchical control: (a) Before layering; (b) After layering

As shown in Table 4, under similar initial and final SOC conditions, the fuel consumption per 100 kilometers is reduced across different reinforcement learning algorithms when the hierarchical control strategy is employed, compared to the non-hierarchical approach. The bar chart in Fig. 8 provides a clearer visualization of these fuel consumption improvements. The figure shows that the hierarchical control strategy resulted in reductions of 7.52 %, 5.29 %, 9.6 %, and 5.93 % in fuel consumption, respectively, when compared to the non-hierarchical case.

As shown in Table 5, with DQN algorithm as an example, the strategy with hierarchical control resulted in reductions of 18.3 %, 14.23 %, 16.94 %, and 20.9 % in the emissions of various harmful pollutants such as *CO*, *HC*, *PM*, NO_x, compared to the non-hierarchical approach.

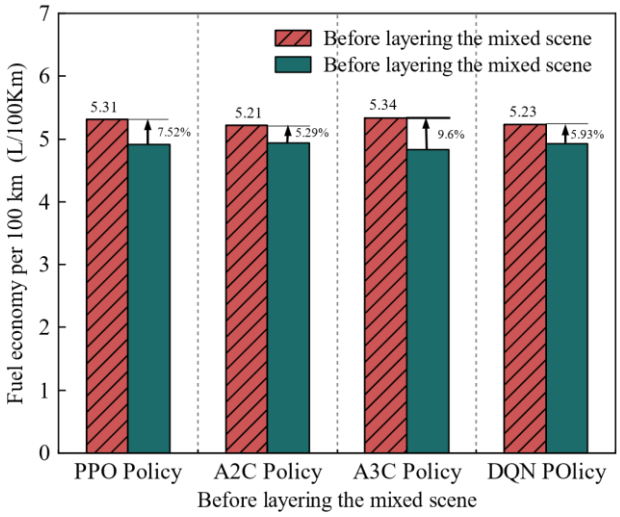


Fig. 8 Fuel consumption comparison for different algorithms

Table 4 Fuel consumption per 100 km for different algorithms with similar initial and final SOC

Strategy	Initial/Final SOC	Fuel consumption per 100 km (L/00 Km)	Depletion rate
PPO SHEMS (offline)	0.65/0.51	5.314	
PPO SHEMS (HIL)	0.65/0.51	4.917	7.52 %
A2C SHEMS (offline)	0.65/0.51	5.212	
A2C SHEMS (HIL)	0.65/0.502	4.936	5.29 %
A3C SHEMS (offline)	0.65/0.514	5.34	
A3C SHEMS (HIL)	0.65/0.51	4.826	9.6 %
DQN SHEMS (offline)	0.65/0.489	5.23	
DQN SHEMS (HIL)	0.65/0.495	4.92	5.93 %

Table 5 Pollutant emissions with the DQN algorithm

Pollutant	DQN strategy (Before layering)	DQN strategy (After layering)	Decrement rate Before and after layering	Dynamic programming
CO	10.25	8.34	18.3 %	8.24
HC	2.81	2.41	14.23 %	2.40
PM	0.098	0.0814	16.94 %	0.0809
NO_x	2.68	2.12	20.9 %	2.14

5.3 Hardware-in-the-Loop Experiment

5.3.1 Hardware-in-Loop Experiment

To verify the effect of the strategy in real controllers, we have built a Hardware-in-the-Loop (HIL) platform. As shown in Fig. 9, the experimental system consists of a hybrid power model, a driver operation system, a virtual scenario system, a sensor system, an NI real-time system, and a vehicle control unit. The virtual scenario system provides the driver with a realistic driving environment, making the driving experience more similar to reality. Furthermore, it provides traffic environment information, road information, and geographic location information through data interaction. The main function of the vehicle control system is to implement the proposed strategy and output control parameters to the actuators. The driver's operation information is all fed back to the steering system, while the vehicle's speed status information and the status of the electromagnetical system are provided by the real-time simulation system.

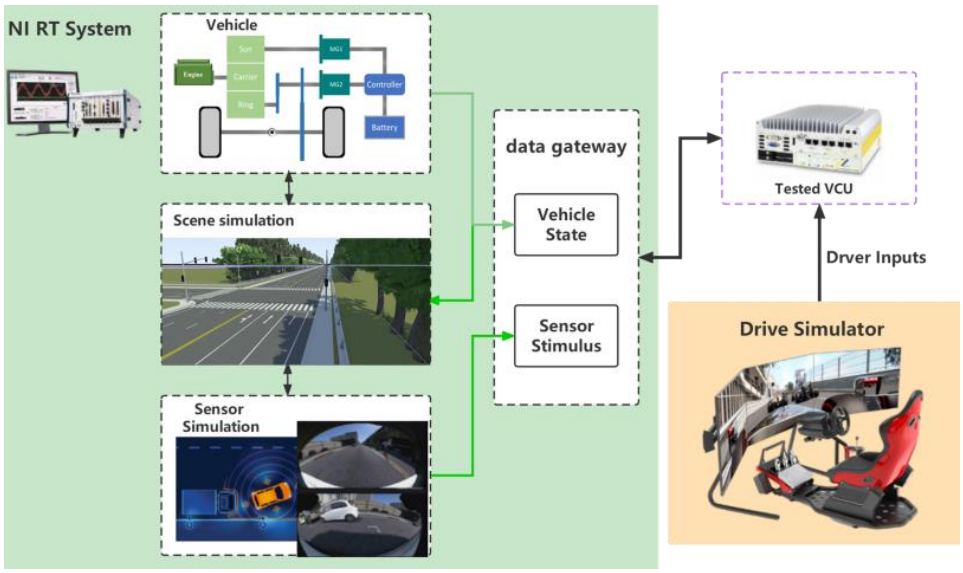


Fig. 9 HIL test bench

The integrated system is shown in Fig. 10(a). In Fig. 10(b), the data acquisition system is combined with the driver operation system, displayed below the driver. Based on the existing configuration and technical conditions, CAN communication technology is applied to achieve data interaction, and real-time data of the steering wheel angle, acceleration, and brake pedal is obtained. These data are input into the Vehicle Control Unit (VCU).

(a) (b)

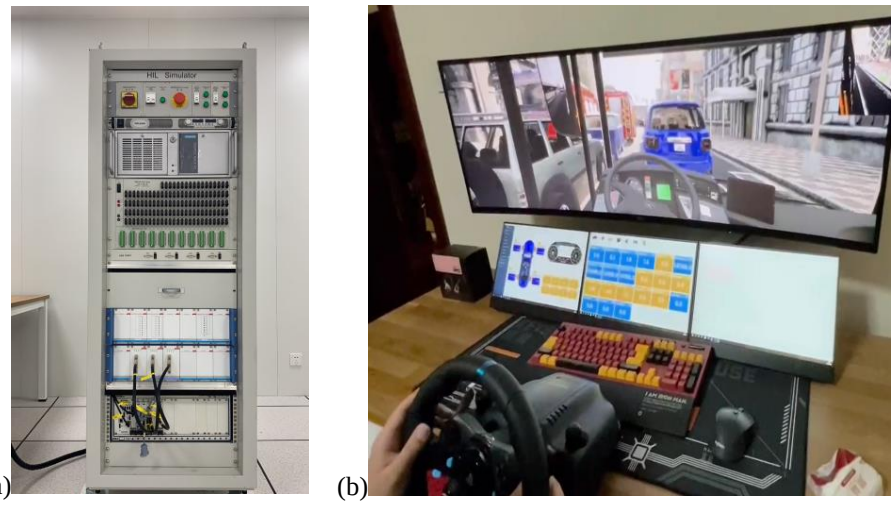


Fig. 10 Integrated system and driver operating system: (a) integrated system; (b) Driver operating system

5.3.2 Hardware-in-the-Loop Experiment Data Analysis

To further validate the hierarchical strategy proposed in this work, Hardware-in-the-Loop (HIL) experiments based on the test driving cycle are conducted. The validation process is composed of three key aspects: verifying the vehicle following algorithm, analyzing the *SHEMS*, and validating the harmful emissions reduction. As shown in Fig. 11, the vehicle following performance can be observed. From the figure, it is evident that the vehicle is able to closely follow the lead vehicle without any collision incidents, demonstrating good driving safety.

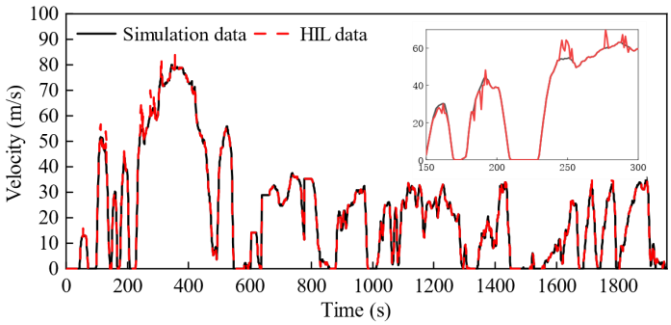


Fig. 11 Performance of vehicle following in HIL

The range of SOC variations during the simulation and HIL tests is shown in Fig. 12. The blue line represents the battery SOC changes under the offline simulation, while the red line shows the battery SOC changes in the HIL test. From the figure, it can be seen that both strategies are able to maintain a good range of battery SOC under the real-time operation. The battery performance and state remain in a favorable condition, indicating that the battery is functioning normally, which ensures the overall reliability and stability of the strategy.

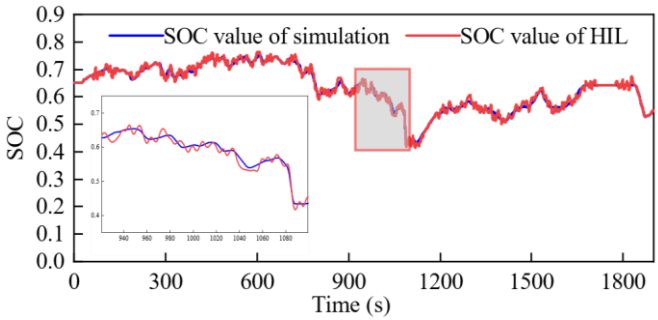
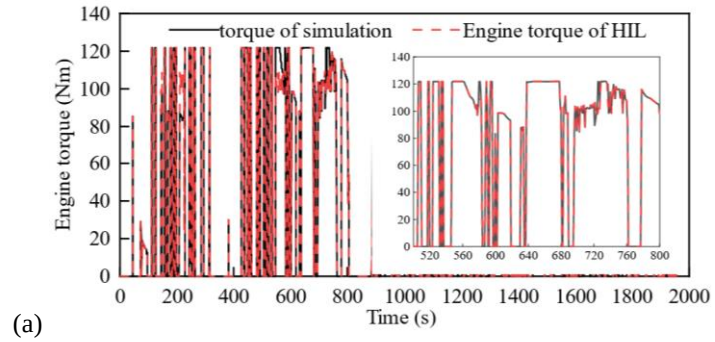


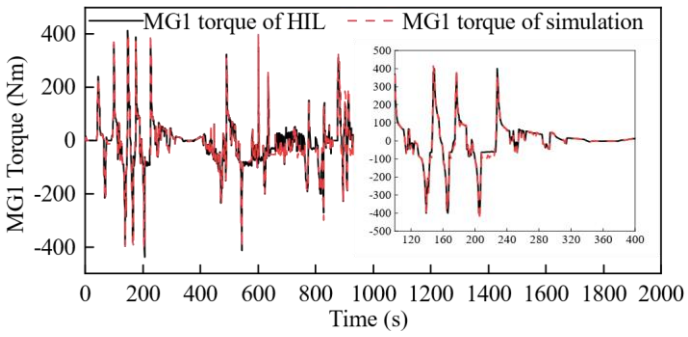
Fig. 12 Performance of battery SOC changes in HIL

The comparative plots of the engine torque, engine MG1 torque, and electric motor MG2 torque under the test driving cycle are depicted in Fig. 13. The inset plot displays the detailed view of the local variations. It can be concluded that there are some differences

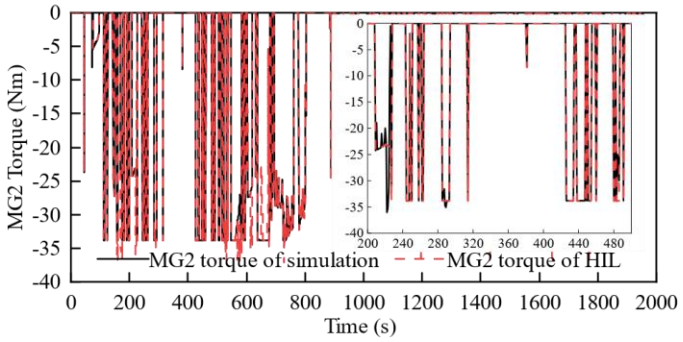
between the HIL test data and the offline simulation data. However, these differences are within a reasonable range. Overall, the variations in the real-time environment between the two sets of data are not significant, and the alignment between them is relatively satisfied.



(a)



(b)



(c)

Fig. 13 Performance in HIL: (a) Engine torque, (b) MG1 torque (c) MG2 torque

As shown in Table 6, by comparing the emissions of harmful substances between simulation and HIL testing using the DQN algorithm, it can be summarized that the layered DQN algorithm significantly reduced pollutant emissions in the HIL experiment, with the emissions approaching the levels achieved under the dynamic programming approach.

Furthermore, the results from the HIL testing closely matched those from the simulation, with only minor differences.

Table 6 Comparison of emissions of harmful substances between simulation and HIL under DQN algorithm

Harmful gasses	DQN (before layering) Simulation/HIL	DQN (after layering) Simulation/HIL	Dynamic programming Simulation/HIL
CO	10.25/10.02	8.43/8.31	8.24/8.23
HC	2.81/2.95	2.41/2.40	2.40/2.40
PM	0.098/0.094	0.0814/0.0801	0.0809/0.0810
NO_x	2.68/2.67	2.12/2.13	2.14/2.13

6. CONCLUSION

The *SHEMS* refers to driving techniques and strategies that aim to reduce the environmental impact of vehicles by improving vehicle operation and driving behavior. Based on a learning-based approach, a layered ecological driving strategy targeting a hybrid electric vehicle is proposed, with the objectives of driving safety, energy management, and pollutant emission reduction. Simulation and HIL testing are conducted to evaluate the proposed ecological driving strategy, and the performance is assessed. The main research work of this paper is as follows:

The driving scenarios for the *SHEMS* can be classified into free driving, car-following, and intersection scenarios and the optimization objectives are formulated for each scenario.

To account for the impact of traffic conditions and driver behavior on driving strategies, an imitation learning–based approach is introduced as an innovative solution. This strategy constructs an adaptive learning framework, which not only reduces the complexity of reward function design, but also enables the model to exceed expert demonstrations and capture high-dimensional environmental features. The simulation verification shows that the algorithm performs excellently in car-following, vehicle distance maintenance, and speed regulation. The average differences between the expert algorithm and the proposed algorithm are 0.38 m/s, 0.0026 m/s, and 0.21 m/s under steady-state, acceleration, and deceleration conditions, respectively.

For the complex intersection scenario, two strategies are proposed creatively, which consists of incorporating an additional reward mechanism and a self-supervised reinforcement learning module to effectively compensate for the sparsity of the ecological driving reward function. To address the challenges posed by the changing intersection environment, a reinforcement learning calibration strategy is introduced to correct the algorithm, significantly improving its efficiency. The reward function not only converges rapidly but also has a denser reward distribution. Based on the simulation results, the strategy proposed in this paper has significant advantages in terms of reward value, battery SOC, and engine operation. Under different algorithms, the fuel consumption is significantly reduced through the layered strategy, with reductions of 7.52 %, 5.29 %, 9.6%, and 5.93 % for the PPO, A2C, A3C, and DQN strategies, respectively. Taking the DQN algorithm as an example, the emissions of harmful gases *CO*, *HC*, *PM* and NO_x, are

reduced by 18.3 %, 14.23 %, 16.94 %, and 20.9 %, respectively, after layering. The HIL experiment further verifies the safety and effectiveness of the strategy, demonstrating the algorithm's efficiency in reducing harmful gas emissions.

Acknowledgement: *This work was supported by the Guangxi Science and Technology Major Program (Grant No. Guike AA23062067), in part by the Open Fund for State Key Laboratory of Advanced Design and Manufacturing Technology for Vehicle at Hunan University under Grant 82315002.*

REFERENCES

1. Huang, Y., Ng, E., Zhou, J., Surawski, N., Lu, X., Du, B., Forehead, H., Perez, P., Chan, E., 2021, *Impact of drivers on real-driving fuel consumption and emissions performance*, Science of the Total Environment, 798, 149297.
2. Li, J., Fotouhi, A., Liu, Y., Zhang, Y.,Chen, Z., 2024, *Review on eco-driving control for connected and automated vehicles*, Renewable & Sustainable Energy Reviews, 189, 114025.
3. Ahmad, A., Al-Sumaiti, A., Byon, Y.,Al Hosani, K., 2024, *Eco-Driving Framework for Autonomous Vehicles at Signalized Intersection in Mixed-Traffic Environment*, IEEE Access, 12, pp. 85291-85305.
4. Yuan, W., Frey, H., 2020, *Potential for metro rail energy savings and emissions reduction via eco-driving*, Applied Energy, 268, 114944.
5. Wu, F., Ye, H., Bektas, T., Dong, M., 2025, *New and tractable formulations for the eco-driving and the eco-routing-and-driving problems*, European Journal of Operational Research, 321(2), pp. 445-461.
6. Yu, M., Long, J., 2022, *An Eco-Driving Strategy for Partially Connected Automated Vehicles at a Signalized Intersection*, IEEE Transactions on Intelligent Transportation Systems, 23(9), pp. 15780-15793.
7. Guo, Q., Angah, O., Liu, Z., Ban, X., 2021, *Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors*, Transportation Research Part C-Emerging Technologies, 124, 102980.
8. He, W., Huang, Y., 2021, *Real-time Energy Optimization of Hybrid Electric Vehicle in Connected Environment Based on Deep Reinforcement Learning*, 6th IFAC Conference on Engine Powertrain Control, Simulation and Modeling (E-COSM), 54, pp. 176-181.
9. Tadić, D., Lukić, J., Komatina, N., Marinković, D., Pamučar, D., 2025, *A Fuzzy Decision-Making Approach to Electric Vehicle Evaluation and Ranking*, *Tehnicki Vjesnik*, 32(3), pp. 1066-1075.
10. Hu, J., Shao Y., Sun, Z., Wang M., Bared, J., Huang, P., 2016, *Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication*, Transportation Research: Part C, 68 pp. 228-244.
11. Liu, L., Neng, L., Peng, Z., Zhan, S., Gao, J., Wang, H., 2024, *Modeling and Verification of Eco-Driving Evaluation*, Journal of Information Processing Systems, 20(3), pp. 296-306.
12. Li, J., Fotouhi, A., Pan, W., Liu, Y., Zhang, Y., Chen, Z., 2023, *Deep reinforcement learning-based eco-driving control for connected electric vehicles at signalized intersections considering traffic uncertainties*, Energy, 279, 128139.
13. Bai, Z., Hao, P., Shangguan, W., Cai, B., Barth, M., 2022, *Hybrid Reinforcement Learning-Based Eco-Driving Strategy for Connected and Automated Vehicles at Signalized Intersections*, IEEE Transactions on Intelligent Transportation Systems, 23(9), pp. 15850-15863.
14. Xie, S., Hu, X., Liu, T., Qi, S., Lang, K., Li, H., 2019, *Predictive vehicle-following power management for plug-in hybrid electric vehicles*, Energy, 166, pp. 701-714.
15. Qi, C., Zhu, Y., Song, C., Yan, G., Wang, D., Xiao, F., Zhang, X., Cao, J., Song, S., 2022, *Hierarchical reinforcement learning based energy management strategy for hybrid electric vehicle*, Energy, 238, 121703.
16. Marinković, D., Dezső, G., Milojević, S., 2024, *Application of machine learning during maintenance and exploitation of electric vehicles*, Advanced Engineering Letters, 3(3), pp. 132-140.
17. Yeom, K., 2022, *Model predictive control and deep reinforcement learning based energy efficient eco-driving for battery electric vehicles*, Energy Reports, 8, pp. 34-42.
18. He, T., Liang, C., Zheng, C., Liu, Y., Zhang, Y., Hu, J., 2025, *Multi-Objective Autonomous Eco-Driving Strategy: A Pathway to Future Green Mobility*, Green Energy and Intelligent Transportation, 4(4), 100279.

19. Ozatay, E., Onori, S., Wollaeger, J., Ozguner, U., Rizzoni, G., Filev, D., Michelini, J., Di Cairano, S., 2014, *Cloud-Based Velocity Profile Optimization for Everyday Driving: A Dynamic-Programming-Based Solution*, IEEE Transactions on Intelligent Transportation Systems, 15(6), pp. 2491-2505.
20. Wang, S., Lin, X., 2020, *Eco-driving control of connected and automated hybrid vehicles in mixed driving scenarios*, Applied Energy, 271, 115233.
21. Guo, L., Zhang, X., Zou, Y., Han, L., Du, G., Guo, N., Xiang, C., 2022, *Co-optimization strategy of unmanned hybrid electric tracked vehicle combining eco-driving and simultaneous energy management*, Energy, 246, 123309.
22. Liu, Y., Huang, B., Yang, Y., Lei, Z., Zhang, Y., Chen, Z., 2022, *Hierarchical speed planning and energy management for autonomous plug-in hybrid electric vehicle in vehicle-following environment*, Energy, 260, 125212.
23. Peng, J., Zhang, F., Coskun, S., Hu, X., Yang, Y., Langari, R., He, J., 2023, *Hierarchical Optimization of Speed Planning and Energy Management for Connected Hybrid Electric Vehicles Under Multi-Lane and Signal Lights Aware Scenarios*, IEEE Transactions on Intelligent Transportation Systems, 24(12), pp. 14174-14188.
24. Li, J., Liu, Y., Zhang, Y., Lei, Z., Chen, Z., Li, G., 2021, *Data-driven based eco-driving control for plug-in hybrid electric vehicles*, Journal of Power Sources, 498, 229916.
25. Qiu, S., Qiu, L., Qian, L., Pierluigi, P., 2019, *Hierarchical energy management control strategies for connected hybrid electric vehicles considering efficiencies feedback*, Simulation Modelling Practice and Theory, 90, pp. 1-15.
26. Almannaa, M., Chen, H., Rakha, H., Loulizi, A., El-Shawarby, I., 2019, *Field implementation and testing of an automated eco-cooperative adaptive cruise control system in the vicinity of signalized intersections*, Transportation Research Part D-Transport and Environment, 67, pp. 244-262.
27. Zheng, Y., Guo, R., Ma, D., Zhao, Z., Li, X., 2020, *A Novel Approach to Coordinating Green Wave System with Adaptation Evolutionary Strategy*, IEEE Access, 8, pp. 214115-214127.
28. Chen, Z., Xiong, S., Chen, Q., Zhang, Y., Yu, J., Jiang, J., Wu, C., 2022, *Eco-Driving: A Scientometric and Bibliometric Analysis*, IEEE Transactions on Intelligent Transportation Systems, 23(12), pp. 1-21.
29. Ma, H., Xie, H., Brown, D., 2018, *Eco-Driving Assistance System for a Manual Transmission Bus Based on Machine Learning*, IEEE Transactions on Intelligent Transportation Systems, 19(2), pp. 572-581.
30. Chen, H., Rakha, H., Loulizi, A., El-Shawarby, I., Almannaa, M., 2016, *Development and Preliminary Field Testing of an In-Vehicle Eco-Speed Control System in the Vicinity of Signalized Intersections*, 14th IFAC Symposium on Control in Transportation Systems (CTS), 49(3), pp. 249-254.
31. Dong, H., Zhuang, W., Chen, B., Lu, Y., Liu, S., Xu, L., Pi, D., Yin, G., 2022, *Predictive energy-efficient driving strategy design of connected electric vehicle among multiple signalized intersections*, Transportation Research Part C-Emerging Technologies, 137, 103595.
32. Yao, H., Li, X., 2020, *Decentralized control of connected automated vehicle trajectories in mixed traffic at an isolated signalized intersection*, Transportation Research Part C-Emerging Technologies, 121, 102846.
33. Kebria, P., Khosravi, A., Salaken, S., Nahavandi, S., 2020, *Deep imitation learning for autonomous vehicles based on convolutional neural networks*, IEEE-Caa Journal of Automatica Sinica, 7(1), pp. 82-95.
34. Zhang, C., Bai, W., Du, X., Liu, W., Zhou, C., Qian, H., 2023, *Survey of imitation learning : tradition and new advances*, *Journal of Image and Graphics*, 28(6), pp. 1585-1607.
35. Hussein, A., Gaber, M., Elyan, E., Jayne, C., 2017, *Imitation Learning: A Survey of Learning Methods*, ACM Computing Surveys, 50(2), 21.
36. Zhong, Q., Zhi, J., Xu, Y., Gao, P., Feng, S., 2024, *Applying an Extended Theory of Planned Behavior to Predict Young Drivers' In-Vehicle Information System (IVIS) Use Intention and Behavior While Driving: A Longitudinal Two-Wave Survey*, Sustainability, 16(20), 8908.
37. Huang, H., Zeng, Z., Yao, D., Pei, X., Zhang, Y., 2022, *Spatial-temporal ConvLSTM for vehicle driving intention prediction*, Tsinghua Science and Technology, 27(3), pp. 599-609.
38. Emerson, R., 2024, *K-Means Clustering Explained*, Journal of Visual Impairment & Blindness, 118(1), pp. 65-66.
39. Savitzky, A., Golay, M., 1964, *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*, *Analytical Chemistry*, 36(8), pp. 1627-1639.
40. Brockfeld, E., Kühne, R., Wagner, P., 2005, *Calibration and Validation of Microscopic Traffic Flow Models*, Transportation Research Record, 1934, pp. 62-70.