Uncertainty-driven Active Reinforcement Learning for Energy Management Strategy in Electrified Vehicles

Dong Hu, Cheng Tian, Anh-Tu Nguyen, Chao Huang

Uncertainty-driven Active Reinforcement Learning for Energy Management Strategy in Electrified Vehicles

Dong Hu, Cheng Tian, Anh-Tu Nguyen *Senior Member, IEEE,*
and Chao Huang, *Senior Member, IEEE,*

***Abstract*—Reinforcement learning (RL), as a data-driven optimization method, enables electrified vehicles to train optimal energy management strategy (EMS) in complex and dynamic environments. However, traditional RL methods often face inefficiency during the exploration process, particularly in high-dimensional continuous state spaces, leading to slow convergence or a tendency to fall into local optima. To address these issues, this paper proposes an uncertainty-driven active RL (UaRL) for EMS, which is built upon the soft actor-critic algorithm, using the dual Q-values difference as a measure of uncertainty. When the uncertainty increases significantly, the active learning mechanism is activated, allowing the UaRL system to select more reliable actions through rule-based methods and guide the improvement of the RL policy. Experimental results show that the novel UaRL approach has significant advantages in energy efficiency, convergence speed, and policy stability. Compared to vanilla RL, the proposed method improves economic performance by an average of 14.68%. It also demonstrates superior performance in battery health management and capacity stability. This study provides an efficient and reliable hybrid learning framework for EMS in electrified vehicles.**

***Index Terms*—Reinforcement learning, Energy management, Electrified vehicles, Active learning, Uncertainty measurement.**

I. Introduction

GLOBAL energy consumption is rising rapidly, threatening environmental and economic sustainability. Fossil fuels dominate energy use, worsening the climate crisis [1]. In 2023, transport accounted for 23% of global CO_2 emissions, with road transport contributing nearly three-quarters [2]. Electric vehicles (EVs) exceeded 40 million globally, representing 18% of new car sales [3]. Vehicle electrification, particularly hybrid electric vehicles (HEVs), offers a viable solution to cut fuel use and emissions. Effective energy management strategies (EMS) are essential to improving HEV efficiency and reducing environmental impact [4].

Traditional EMS face notable limitations. Rule-based strategies rely on fixed heuristics, making them inflexible under complex driving conditions. Optimization-based methods like dynamic programming (DP) and model predictive control are theoretically sound but often too computationally intensive for real-time use [5]. By interacting with the environment, reinforcement learning (RL) adapts to changing conditions and enables efficient real-time decision-making [6]. When formulated as a multi-objective optimization problem, RL also can balance competing goals and overcome the shortcomings of traditional EMS approaches [7].

However, mainstream RL methods still suffer from low sample efficiency, primarily due to high-dimensional and continuous state-action spaces, which results in extended trial-and-error processes [8]. To alleviate this issue, incorporating expert knowledge has emerged as a promising direction. By enabling interaction between agents and experts, interactive and collaborative machine learning (ML) approaches can guide exploration, and reduce unnecessary trials [9]. Expert feedback can take various forms: some methods allow agents to imitate expert behaviors directly, while others involve evaluating state-action pairs to label them as beneficial or detrimental, thereby shaping the learning process [10], [11].

A third approach integrates experts into the interaction loop between the agent and the environment through interventions [12]. While these methods are promising, challenges remain. For instance, determining the optimal timing and frequency of expert interventions is still an area of active research. Our previous work [13] uses a deep ensemble method to trigger interventions based on the deviation between agent and expert actions, but it does not fully account for the agent's training instability. Moreover, building expert models often requires fitting high-quality expert data, which can significantly increase training costs in practice.

Given these limitations, it is essential to determine when and how to effectively integrate expert guidance into the RL training process. In the early training stage, RL agents often lack reliable knowledge, leading to poor decisions and potential convergence to suboptimal policies. So we propose an uncertainty-driven active RL (UaRL) framework for EMS, which quantifies uncertainty by utilizing the differences between dual Q-networks, reflecting the agent's confidence in its current strategy. When uncertainty is high—indicating low confidence—an active learning (AL) mechanism is triggered to incorporate expert guidance. This helps the agent select more reliable actions, improving policy quality and accelerating convergence, particularly during unstable early training.

This work was supported by PROCORE-France/Hong Kong Joint Research Scheme (F-PolyU501/23). *(Corresponding author: Chao Huang.)*

Dong Hu and Chao Huang are with the Department of Industrial and Systems Engineering, the Hong Kong Polytechnic University, Hong Kong (E-mail: dong24.hu@connect.polyu.hk; hchao.huang@polyu.edu.hk).

Cheng Tian is with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong (e-mail: cheng7.tian@connect.polyu.hk).

Anh-Tu Nguyen is with the LAMIH Laboratory, UMR CNRS 8201, and INSA Hauts-de-France, Université Polytechnique Hauts-de-France, France (e-mail: tnguyen@uphf.fr).

A range-extended electric bus (REEB) is used as the platform to evaluate the proposed method. The main contributions are as follows:

- An active RL framework is proposed to improve the efficiency of RL-based EMS by allowing the agent to selectively query expert advice, reducing exploration cost and enhancing learning efficiency.
- The framework employs critic-network-based uncertainty estimation to trigger active expert intervention using simple prior strategies.
- Extensive testing demonstrates that the proposed method outperforms baseline approaches in training efficiency, stability, energy savings, and generalization.

II. Related work

A. Data-Driven EMS

To tackle the limitations of traditional EMS—including poor real-time performance, and restricted optimization—RL has emerged as a promising alternative [5]. By eliminating the need for precise system modeling and leveraging self-learning, RL demonstrates strong adaptability in complex driving scenarios [14]. Despite notable progress, there are still key challenges in improving training efficiency, stability, and robustness. Model-based RL [15] and transfer-based RL [16] offer efficiency gains, yet model-based methods are constrained by inherent algorithmic limits, and transfer-based approaches are heavily reliant on prior domain features. While offline RL enables policy learning from pre-collected data—such as the advantage-based replay mechanism proposed by Niu et al. [17]—it is hampered by the high cost of quality data acquisition, as well as persistent issues like data mismatch and insufficient exploration.

Recent studies have explored the integration of RL with other techniques. For example, by incorporating driving condition prediction models [18], EMS can predict future driving demands, which enhances its foresight and dynamic adaptability. Predictions of surrounding vehicle motion states [19], traffic interaction behaviors [20], and passenger demands [21] have also been investigated to determine their potential impact on energy management. Chen et al. [22] proposed an integrated framework that optimizes trajectory and energy, enhancing energy efficiency and system health. These methods can partially address EMS limitations, but they often rely heavily on accurate predictive models and external information, making them vulnerable to error propagation and uncertainty. In addition, most RL-EMS approaches lack active reliability awareness, especially in untrained or uncertain environments. Although confidence-aware EMS [23] improves self-awareness by evaluating agent confidence, it usually incurs high computational and data costs.

B. Active Learning

Traditional supervised learning depends heavily on large amounts of labeled data, which is time-consuming and often impractical under limited resources. AL addresses this by improving model performance at lower cost, focusing training on the most informative samples. In pool-based AL, the model selects uncertain samples, queries their labels, and updates accordingly. Common selection criteria include entropy [24], mutual information [25], and change rate [26]. Selective sampling [27] further combines AL with online learning by sequentially deciding whether to query labels.

Integrating AL into RL has shown promising results. Schulze et al. [28] incorporated query costs into the reward function, while Krueger et al. [29] studied cost-sensitive reward observation using heuristic methods. Tucker et al. [30] explored costly rewards in multi-armed bandits, demonstrating empirical gains and achieving a regret bound of $O(T^{2/3})$. A core challenge in AL for RL is designing reliable query mechanisms. Epistemic uncertainty, often measured via Bayesian networks or deep ensembles [6], is a common indicator, though these methods are typically computationally expensive and difficult to train [31]. Compared to traditional methods, AL reduces labeling costs by focusing on critical samples and enhances EMS robustness and generalization under rare conditions. Knowledge-guided ML [12] and data-efficient optimization further enhance reliability, especially when data or experimental access is limited.

III. Energy flow model

This study uses a range-extended medium-sized bus with a series hybrid powertrain, where the engine generates electricity to power the motor, which drives the wheels. The following backward vehicle model is used to represent the required torque $F_{\mathrm{req},t}$ at time step t [32]:

$$F_{\mathrm{req},t} = mgC_f + \frac{1}{2}\varsigma_a A_d C_d v_t^2 + \delta_v m a_t, \tag{1}$$

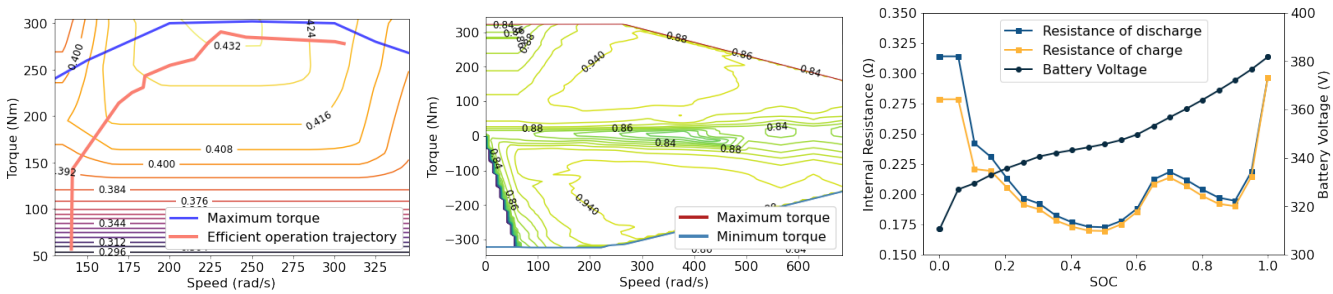
where, m represents the mass of the REEB, g is the gravitational acceleration, C_f is the rolling resistance coefficient, ς_a is the air density, A_d is the frontal area of the vehicle, and C_d is the aerodynamic drag coefficient. Additionally, v_t and a_t denote the vehicle's speed and acceleration, respectively, while δ_v is the conversion factor for the rotational mass. The key parameters of the REEB model are shown in Table I.

TABLE I: Main parameters of the vehicle.

System	Parameters	Value
Vehicle body	Curb weight m	3500 (kg)
	Air resistance coefficient C_d	0.65
	Frontal area A_d	3.9 (m^2)
Engine	Maximum power $P_{\mathrm{eng,max}}$	85 (kW)
	Maximum torque $T_{\mathrm{eng,max}}$	305 (Nm)
Traction motor	Maximum torque $T_{\mathrm{mot,max}}$	320 (Nm)
Generator	Maximum torque $T_{\mathrm{gen,max}}$	277 (Nm)
Battery	Capacity C_n	8.7 (kWh)
	Voltage V	347.8 (V)
Transmission	Final drive ratio i_g	5.857

The REEB is equipped with a range extender for charging lithium-ion battery (LIB), where the universal characteristic map of engine is shown in Fig. 1 (a). The fuel consumption rate $\dot{m}_{f,t}$ of the engine is determined on the basis of its torque and rotational speed, which described by Eq. (2):

$$\dot{m}_{f,t} = f\left(T_{\mathrm{eng},t}, W_{\mathrm{eng},t}\right), \tag{2}$$



(a) (b) (c)

Fig. 1: (a) Universal characteristic map of engine, (b) efficiency map of EM, and (c) parameters of LIB.

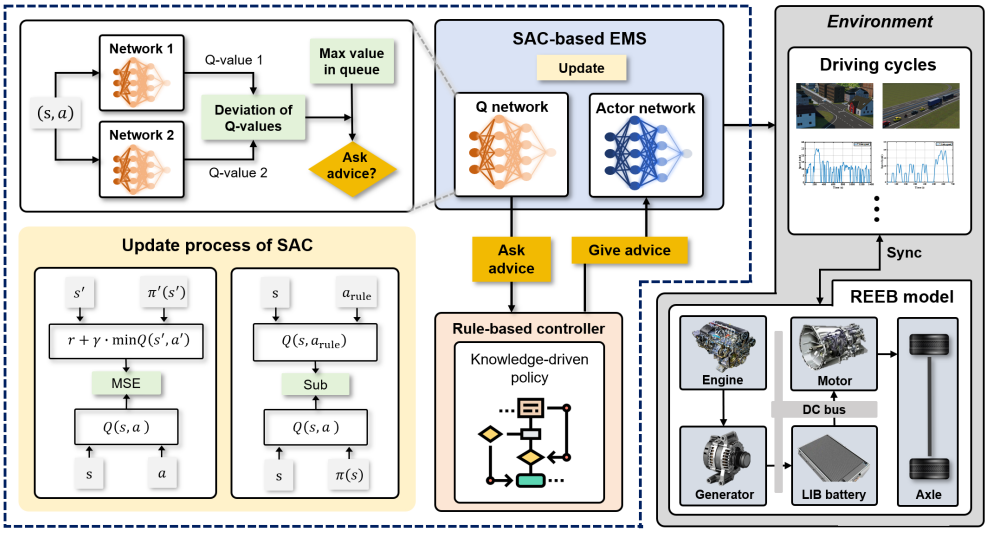


Fig. 2: Overall framework of uncertainty-driven active reinforcement learning.

where $T_{\mathrm{eng},t}$ refers to the engine torque, while $W_{\mathrm{eng},t}$ represents its rotational speed. We have also defined the optimal operating curve for the engine, whose function is to map the engine's output power to the corresponding torque and rotational speed. This mapping reduces the control variables from $[T_{\mathrm{eng},t}, W_{\mathrm{eng},t}]$ to $[P_{\mathrm{eng},t}]$, thereby simplifying the control process.

The electric motor (EM) can function either as a traction or charging unit, requiring the EM model to account for both traction and supplemental energy flows. The torque $T_{\mathrm{mot},t}$ and power $P_{\mathrm{mot},t}$ of EM are expressed as follows:

$$T_{\mathrm{mot},t} = \frac{F_{\mathrm{req},t}}{i_g} \cdot R_{\mathrm{wheel}}, \tag{3}$$

$$P_{\mathrm{mot},t} = \omega_{\mathrm{mot},t} \cdot T_{\mathrm{mot},t} \cdot (\eta_{\mathrm{mot}})^{-\kappa}, \tag{4}$$

where i_g represents the final ratio, R_{wheel} is the wheel radius, $\omega_{\mathrm{mot},t}$ denotes the speed of the EM, and η_{mot} is the electromechanical conversion efficiency, which depends on both $T_{\mathrm{mot},t}$ and $\omega_{\mathrm{mot},t}$. When the traction torque is positive, the coefficient κ is set to 1, and when negative, it becomes -1, indicating regenerative braking. The efficiency map of the EM is shown in Fig. 1 (b).

Fig. 1 (c) shows the charging and discharging internal resistance of LIB and battery voltage. According to the equivalent circuit model, the energy consumption of LIB can be obtained by calculating the current $I_{\mathrm{LIB},t}$ [33]:

$$P_{\mathrm{batt},t} = V_{oc,t} - R_0 I_{\mathrm{LIB},t}^2, \tag{5}$$

$$I_t = \frac{V_{oc,t} - \sqrt{V_{oc,t}^2 - 4 \cdot R_0 \cdot P_{\mathrm{batt},t}}}{2 \cdot R_0}, \tag{6}$$

$$\frac{d\mathrm{SoC}_t}{dt} = -\frac{I_{\mathrm{LIB},t}}{3600 C_n}, \tag{7}$$

where $P_{\mathrm{batt},t}$ denotes the output power, $V_{oc,t}$ represents the open circuit voltage, and R_0 represents the internal resistance. Additionally, C_n refers to the nominal capacity of the LIB.

The cycling current gradually induces degradation in LIB internal materials, which can be described by the dynamic changes in its state of health (SoH), as shown below [34]:

$$\frac{d\mathrm{SoH}_t}{dt} = -\frac{1}{2N(c_t) C_n} \int_0^t |I(\tau)| d\tau, \tag{8}$$

where, N denotes the equivalent number of cycles before the end of its lifespan, and c_t represents the c-rate that is proportional to the current $I_{\mathrm{LIB},t}$ (shown in Table II). Discretizing Eq. (8) with a time step Δt, it follows that:

$$\Delta\mathrm{SoH}_t = -\frac{|I_{\mathrm{LIB},t}| \Delta t}{2N(c_t) C_n}, \tag{9}$$

TABLE II: Dependence of pre-exponential factor to c-rate.

c-rate	0.5	2	6	10
B(c)	31630	21681	12934	15512

where Δt denotes time step duration. The capacity loss percentage, ΔC_n calculated using the Arrhenius equation, is expressed as follows [35]:

$$\Delta C_n = B\left(c_t\right) \cdot e^{\frac{-E_a\left(c_t\right)}{R_g T_a}} \cdot Ah\left(c_t\right)^z, \tag{10}$$

where the activation energy $E_a = (31700 - 370.3 \cdot c_t)$, and the pre-factor B can be calculated according to Table II. T_a is the average temperature of in the LIB system, R_g denotes the ideal gas constant, Ah is the amp-hour throughput, and z is the power-law factor equals 0.55.

The LIB reaches its end of life, when the C_n reduces by 20% (i.e., $\Delta C_n = 20\%$). Based on this condition, the values of Ah and N can be derived using Eq. (10):

$$Ah\left(c_t\right) = \left[20 / \left(B\left(c_t\right) \exp\left(\frac{-E_a\left(c_t\right)}{R_g T_a}\right)\right)\right]^{\frac{1}{2}}, \tag{11}$$

$$N\left(c_t\right) = 3600 Ah\left(c_t\right) / C_n. \tag{12}$$

Remark 1: The battery aging model adopted is grounded in the Arrhenius-based electrochemical formulation for long-term capacity loss estimation (see Eq. (10)), while a cycle-counting approach is used to facilitate real-time control and reward calculation. This hybrid strategy ensures both physical consistency and computational tractability.

IV. Uncertainty-driven active reinforcement learning

As illustrated in Fig. 2, the UaRL framework integrates an soft actor-critic (SAC)-based agent that interacts with the environment in real time to collect experience and optimize its policy. Two Q-values are computed for each state-action pair using a dual Q-network, and their deviation is used to assess uncertainty. When this deviation exceeds a historical threshold, the AL mechanism is triggered. The agent then queries a rule-based controller for a recommended action as a reference during policy update. Additionally, the controller's Q-value for the recommended action is incorporated to further improve policy robustness.

A. Energy Management Formulation

In RL-EMS development, the state captures the system's operating conditions at a given time, enabling the agent to perceive the environment and make informed decisions, while the action represents the agent's control output. Taking external characteristics of the vehicle and its internal powertrain system into account, the vehicle speed v_t, acceleration a_t, and SoC_t are chosen as state variables $\mathbf{s}_t$, while the engine output power $P_{\mathrm{eng},t}$ is designated as the action variable $\mathbf{a}_t$:

$$\begin{cases} \mathbf{s}_t = [v_t, a_t, \mathrm{SoC}_t] \\ \mathbf{a}_t = [P_{\mathrm{eng},t}]. \end{cases} \tag{13}$$

This study focuses on multiple optimization objectives, including energy consumption, SoC regulation, and the health level of LIB. The goal is to minimize the overall expenditure cost, and maintain the SoC within a certain range. Therefore, the reward function consists of the following three parts:

$$r_{\mathrm{fuel},t} = -\omega_1 \cdot \dot{m}_{\mathrm{fuel},t}, \tag{14}$$

$$r_{\mathrm{SoH},t} = -\omega_2 \cdot C_n \Delta \mathrm{SoH}_t, \tag{15}$$

$$r_{\mathrm{SoC},t} = -\omega_3 \cdot \left\|\mathrm{SoC}_t - \mathrm{SoC}_{\mathrm{tar}}\right\|_2^2, \tag{16}$$

where, $r_{\mathrm{fuel},t}$ indicates instantaneous fuel consumption, ω_1 depends on the fuel price (1.02 \$/L). $r_{\mathrm{SoH},t}$ represents the cost related to battery aging, and ω_2 depends on the cost of replacement the LIB packs (139.8 \$/kWh). $r_{\mathrm{SoC},t}$ is used to maintain the battery capacity. $\mathrm{SoC}_{\mathrm{tar}}$ is the target value (0.5) and ω_3 is fixed weight. Therefore, the total reward can be summarized in the following form:

$$r_t = r_{\mathrm{fuel},t} + r_{\mathrm{SoH},t} + r_{\mathrm{SoC},t}. \tag{17}$$

B. Soft Actor-Critic Algorithm

The Markov decision process models RL, where the agent observes state $\mathbf{s}_t$ and selects action $\mathbf{a}_t$ via policy π. The environment then returns reward r_t and transitions the agent to state $\mathbf{s}_{t+1}$. SAC uses temporal difference (TD) methods to update the critic network Q, which guides the optimization of the policy network π [36].

The optimization objective of maximum entropy RL explicitly includes a regularization term for policy entropy, encouraging exploration and preventing premature convergence. The objective is formulated as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_t r_t + \beta H\left(\pi(\cdot \mid \mathbf{s}_t)\right)\right], \tag{18}$$

where $\mathbb{E}_{\pi}$ denotes the expectation over trajectories induced by policy π. For a continuous action space, the entropy $H\left(\pi(\cdot \mid \mathbf{s}_t)\right)$ of the action distribution at state $\mathbf{s}_t$ is calculated as:

$$H\left(\pi(\cdot \mid \mathbf{s}_t)\right) = -\int_{\mathcal{A}} \pi(\mathbf{a} \mid \mathbf{s}_t) \log \pi(\mathbf{a} \mid \mathbf{s}_t)\, d\mathbf{a}, \tag{19}$$

where $\mathcal{A}$ is the continuous action space. In practical implementations, this entropy is estimated by sampling actions from the policy and computing $-\log \pi(\mathbf{a}_t \mid \mathbf{s}_t)$ at each time step. The coefficient β serves as a regularization parameter that controls the trade-off between maximizing expected reward and entropy. To automatically adjust β, the following loss is optimized:

$$\mathcal{L}^{\beta} = \mathbb{E}\left[-\beta \log \pi(\mathbf{a}_t \mid \mathbf{s}_t) - \beta \mathcal{H}_0\right], \tag{20}$$

where $\mathcal{H}_0$ is the target entropy, representing the desired randomness in the policy.

The value network Q can be expressed as:

$$Q\left(\mathbf{s}_t, \mathbf{a}_t\right) = \mathbb{E}\left[r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \gamma \mathbb{E}_{s_{t+1}}\left[V\left(\mathbf{s}_{t+1}\right)\right]\right], \tag{21}$$

$$V\left(\mathbf{s}_t\right) = \mathbb{E}\left[Q\left(\mathbf{s}_t, \mathbf{a}_t\right) - \beta \log \pi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)\right], \tag{22}$$

where γ is the discount factor. SAC uses dual Q-networks to suppress overestimation. The optimization objective of Q-networks is defined as:

$$\mathcal{L}^{Q} = \mathbb{E}\big[Q\left(\mathbf{s}_t, \mathbf{a}_t\right) - \big[r_t + \gamma\big(Q\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}\right) - \beta H\left[\pi\left(\mathbf{s}_{t+1}\right)\right]\big)\big]\big]. \tag{23}$$

The policy π is updated as follows:

$$\mathcal{L}^{\pi} = \mathbb{E}\left[\beta \log\left(\pi\left(\mathcal{G}(\mathbf{s}_t) \mid \mathbf{s}_t\right)\right) - Q\left(\mathbf{s}_t, \mathcal{G}(\mathbf{s}_t)\right)\right], \tag{24}$$

where $\mathcal{G}$ is a Gaussian distribution that can achieve efficient optimization and stable training through reparameterization techniques.

C. Rule-Based EMS

We have chosen a hysteresis control-based strategy as the rule-based expert model [23]. Its main purpose is to pursue high efficiency by stabilizing the capacity of LIB. Its value depends on a specific SoC range, and at the beginning step ($t = 0$), its control flag $\mathcal{F}$ can be represented as:

$$\mathcal{F}_0 = 0 \cdot \left[\mathrm{SoC}_0 > \mathrm{SoC}_{\mathrm{tar}}\right] + 1 \cdot \left[\mathrm{SoC}_0 \leq \mathrm{SoC}_{\mathrm{tar}}\right]. \tag{25}$$

Note that $\mathcal{F}_0$ reflects the acceptable deviation between SoC and the target value $\mathrm{SoC}_{\mathrm{tar}}$. Subsequently, with a deviation threshold ϵ, updating $\mathcal{F}$ within the constrained range of SoC, can be expressed as:

$$\mathcal{F}_t = \begin{cases} 0, & \text{if } \mathrm{SoC}_t > \mathrm{SoC}_{\mathrm{tar}} + \epsilon \\ 1, & \text{if } \mathrm{SoC}_t \leq \mathrm{SoC}_{\mathrm{tar}} - \epsilon, \end{cases} \tag{26}$$

where, the activation of this flag indicates that the engine needs to charge LIB at this time.

The basic principle of this rule-based EMS is that the engine operates in its high-efficiency range when the required torque is low. However, if the power does not meet the charging requirements of LIB, that is, when the SoC is below the threshold, the engine will increase the power to meet the charging requirements. This process is represented as follows:

$$P_{\mathrm{eng},t}^{\mathrm{Rule}} = \left\lfloor \left\lceil P_{\mathrm{opt}}, K_{\mathrm{con}}\left(\mathrm{SoC}_{\mathrm{tar}} - \mathrm{SoC}_t + K_{\mathrm{urg}}\right)\right\rceil, P_{\mathrm{eng,max}} \right\rfloor, \tag{27}$$

where $P_{\mathrm{eng},t}^{\mathrm{rule}}$ is the engine power determined by the rule-based strategy, P_{opt} is the optimal engine power, and $P_{\mathrm{eng,max}}$ is the maximum engine power. Symbols $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ denote the floor and ceiling functions, respectively. K_{con} is the conversion factor of engine power, while the coefficient K_{urg} affects the response time of the hysteresis algorithm. Both parameters are unitless and manually chosen.

The deactivation of $\mathcal{F}_t$ indicates that the LIB has sufficient energy for traction, and thus the engine system is switched to an idle state. The rule-based policy $\pi^{\mathrm{Rule}}(\mathbf{s}_t)$ determines the engine power at time t based on the current state $\mathbf{s}_t$, and is defined as:

$$\pi^{\mathrm{Rule}}\left(\mathbf{s}_t\right) = \begin{cases} P_{\mathrm{eng},t}^{\mathrm{Rule}}, & \text{if } \mathcal{F}_t = 1 \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

where $P_{\mathrm{eng},t}^{\mathrm{Rule}}$ is the engine power recommended by the rule-based policy (in kW), and $\mathcal{F}_t \in \{0, 1\}$ is a binary indicator representing whether the engine needs to provide traction power ($\mathcal{F}_t = 1$) or not ($\mathcal{F}_t = 0$).

Although the hysteresis-based rule is simple, it provides interpretable guidance for the RL agent, especially during early exploration. Our experiments show that this expert rule is sufficient to improve initial learning across different driving cycles. Moreover, the proposed framework is flexible and can incorporate more advanced expert systems or human feedback as needed.

D. Action Query Strategy

The AL strategy serves as the core component of UaRL, with its primary focus on determining when the agent should request external guidance. To achieve this, we developed an active query strategy based on evaluation uncertainty. Specifically, the agent begins by calculating the difference between the dual Q-networks for the given state-action pair, as follows:

$$D(\mathbf{s}_t, \mathbf{a}_t) = \left|Q_1(\mathbf{s}_t, \mathbf{a}_t) - Q_2(\mathbf{s}_t, \mathbf{a}_t)\right|, \tag{29}$$

where $Q_i(\mathbf{s}_t, \mathbf{a}_t)$, $i = 1, 2$ represents the expected return of the current critic network, while $D(\mathbf{s}_t, \mathbf{a}_t)$ denotes the difference between the estimated values of the current critic networks. Due to significant differences in the distribution of $D(\mathbf{s}_t, \mathbf{a}_t)$ across various tasks, setting a fixed threshold may result in it being unsuitable for specific tasks or driving conditions. Therefore, a sliding window approach is adopted, where the most recent $D(\mathbf{s}_t, \mathbf{a}_t)$ values are stored in a fixed-length M queue Que_m. When $D(\mathbf{s}_t, \mathbf{a}_t)$ meets the following condition:

$$D(\mathbf{s}_t, \mathbf{a}_t) > \max\left(\mathrm{Que}_m\right). \tag{30}$$

Note that if $D(\mathbf{s}_t, \mathbf{a}_t)$ exceeds the maximum value of the elements in the queue, the agent will request action advice. Once the queue length exceeds the predefined limit $M = 20$, the element at the front of the queue is removed. As shown in Fig. 2, $D(\mathbf{s}_t, \mathbf{a}_t)$ indicates the uncertainty in the critic network's estimation. A larger $D(\mathbf{s}_t, \mathbf{a}_t)$ suggests that the agent is unfamiliar with the current state, and continuing exploration may waste resources. In such cases, seeking advice can help the agent adapt more quickly. Additionally, to maintain exploration, the rule-based strategy only proposes advice in the early stages of training as follows:

$$R_{\mathrm{episode}} < D_R, \tag{31}$$

where, the cumulative reward of an episode is represented by R_{episode}, and the query threshold is given by $D_R = R_{\max}/th$, where $R_{\max}$ is the maximum cumulative reward set based on tasks, and $th = 5$ is an manual parameter determining when the rule-based strategy intervenes or stops providing advice.

Remark 2: The dual Q-value difference $|Q_1 - Q_2|$ is used as the uncertainty metric due to its compatibility with the SAC framework, where twin Q-networks naturally capture value divergence. Unlike ensemble or Bayesian methods, this approach leverages existing structures with minimal computational overhead, making it suitable for real-time deployment. Aligned with EMS goals, it directly reflects decision reliability by linking uncertainty to practical outcomes.

E. Active Learning Process

When the agent requests advice, the rule-based policy fully takes over the agent's actions. The overall control action at time t is given by:

$$\mathbf{a}_t = (1 - \Xi_t) \cdot \mathbf{a}_t^{\mathrm{RL}} + \Xi_t \cdot \mathbf{a}_t^{\mathrm{Rule}}, \tag{32}$$

where $\mathbf{a}_t^{\mathrm{RL}}$ is agent's action, and $\mathbf{a}_t^{\mathrm{Rule}}$ is rule-based suggestion action. $\Xi_t \in \{0, 1\}$ is a binary variable indicating whether advice is requested ($\Xi_t = 1$) or not ($\Xi_t = 0$). To ensure consistency, each expert intervention lasts for at least 10 time steps.

In addition, rule-based advice is also stored as tuples $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}, \Xi_t)$ in the experience replay buffer, which can help the agent with training updates. Due to the fact that the benefits of strategic networks depend on the evaluation of critic networks, we designed an advantage function to improve the critic's update process. Before the training maturity of the critic network $Q(\mathbf{s}_t, \mathbf{a}_t)$, the rule-based actions $\mathbf{a}_t^{\mathrm{Rule}}$ usually produce higher Q-value than untrained policy π, i.e., $Q(\mathbf{s}_t, \mathbf{a}_t^{\mathrm{Rule}}) > Q(\mathbf{s}_t, \mathbf{a}_t)$. Therefore, the advantage function $A(\mathbf{s}_t, \mathbf{a}_t)$ is given by the following equation:

$$A(\mathbf{s}_t, \mathbf{a}_t) = Q_i\left(\mathbf{s}_t, \mathbf{a}_t^{\mathrm{Rule}}\right) - Q_i(\mathbf{s}_t, \pi(\mathbf{s}_t)). \tag{33}$$

To comprehensively utilize the requested advices, the training objective of RL needs to be improved when sampling the advice tuples of data-driven strategy. The update process of Q network in Eq.(23) can be rewritten as:

$$\tilde{\mathcal{L}}^Q = \mathcal{L}^Q + \mathbb{E}\left[A_1(\mathbf{s}_t, \mathbf{a}_t)\right], \tag{34}$$

where $A_1 = A$. Then, the learning objective of the actor network in Eq. (24) needs to be adjusted accordingly to:

$$\begin{aligned}\mathcal{L}^\pi = \mathbb{E}\big[&\beta \log\left(\pi\left(\mathcal{G}(\mathbf{s}_t) \mid \mathbf{s}_t\right)\right) - Q\left(\mathbf{s}_t, \mathcal{G}(\mathbf{s}_t)\right) \\ &+\omega_I\left(\mathbf{a}_t^{\mathrm{Rule}} - \mathbf{a}_t^{\mathrm{RL}}\right)\big],\end{aligned} \tag{35}$$

where the imitation weight ω_I is initialized to 0.5 and decays exponentially as 0.99^{epoch} during training, so that imitation learning plays a smaller role over time.

The prioritized experience replay (PER) can evaluate the priority of each tuple and more efficiently sample data. The priority ρ is defined as:

$$\rho_t = \left|\Delta_t^{\mathrm{TD}}\right| + \epsilon, \tag{36}$$

where, ϵ is a small positive constant, and Δ_t^{TD} is calculated by:

$$\Delta_t^{\mathrm{TD}} = r_t + \gamma \cdot Q\left(\mathbf{s}_{t+1}, \pi\left(\mathbf{s}_{t+1}\right)\right) - Q\left(\mathbf{s}_t, \mathbf{a}_t\right). \tag{37}$$

To better sample request advice, we optimized the priority of tuples using Q-advantage function:

$$\hat{\rho}_t \equiv \rho_t + \Xi_t \cdot A_2, \tag{38}$$

where $A_2 = \exp\left[A(\mathbf{s}_t, \mathbf{a}_t)\right]$, the exponential design further amplifies the differences between trajectories and assigns higher weights to the trajectories of experts in the PER mechanism.

Algorithm 1 describes the general process of UaRL training, and Table III shows the hyperparameter design of the proposed framework's network structure. During the testing phase, the critic network is still running, evaluating the direct gap between dual Q-value to determine whether to ask advice.

Algorithm 1 Uncertainty-driven active reinforcement learning

Require: Maximum episode number N_{ep}, experience buffer $\mathcal{D}$, the critic networks Q_1, Q_2.
Ensure: Trained policy π.

```
for i = 1:N_ep do
    Step j ← 0;
    Observe the initial state s_1;
    while not done do
        Select action a_t^RL = π(· | s_t) ;
        Compute D(s_t, a_t) using Eq. (29)
        if ask advice then according to Eq. (30) and (31)
            Execute expert action a_t = a_t^Rule;
        else
            Execute a_t^RL ;
        end if
        update step j ← j + 1;
        Get s_{t+1}, r_t and store (s_t, a_t, r_t, s_{t+1}, Ξ_t) to D;
        Sample a batch tuple from D;
        Update priority using Eq. (38);
        Update critic networks using Eq. (34);
        if j mod 2 == 0 then
            Update actor network using Eq. (35);
        end if
    end while
end for
```

TABLE III: Network settings and hyperparameters of the proposed EMS.

Component/Parameter	Setting	Value
Network Architecture		
Actor	FC layers	[128, 64, 32] (ReLU)
	Output head	FC (32→ n_a), Sigmoid
	Log std head	FC (32→ n_a)
Critic (Twin Q)	FC layers	[128, 64, 32] (ReLU)
	Output	FC (32→ n_a), linear
Training Hyperparameters		
Training episodes	N_{ep}	300
Replay buffer size	D_m	5×10^4
Batch size	B_a	64
Discount factor	γ	0.99
Soft update coeff.	τ	1×10^{-3}
Actor/Critic LR	lr_A / lr_C	1×10^{-3}
Entropy LR	lr_β	1×10^{-4}
Initial temperature	β_{init}	0.05
Policy update freq.	N_{freq}	2
Rule-based EMS	ϵ	0.03
	K_{con}	1×10^6
	K_{urg}	0.08

Note: n_a denotes the dimensions and action. All FC layers are fully connected. LR denotes learning rate.

V. Results and Discussion

This section provides a detailed description of the experimental design and result analysis to demonstrate the effectiveness of the proposed UaRL for EMS.

TABLE IV: Details of real-world driving cycles.

Cycles	Duration (s)	Velocity (m/s)	Acceleration (m/s^2)	Mileage (km)
V01	1766 (Idle: 174)	12.84±6.44 (Max: 21.33)	0.26±0.34 (Max: 3.33)	22.67
V02	2401 (Idle: 470)	8.01±6.43 (Max: 20.42)	0.29±0.37 (Max: 2.23)	19.24

Note: Acceleration denotes its absolute value.

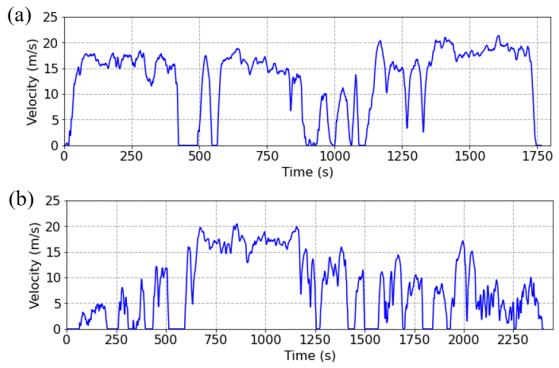


Fig. 3: Driving profiles for (a) V01, and (b) V02.

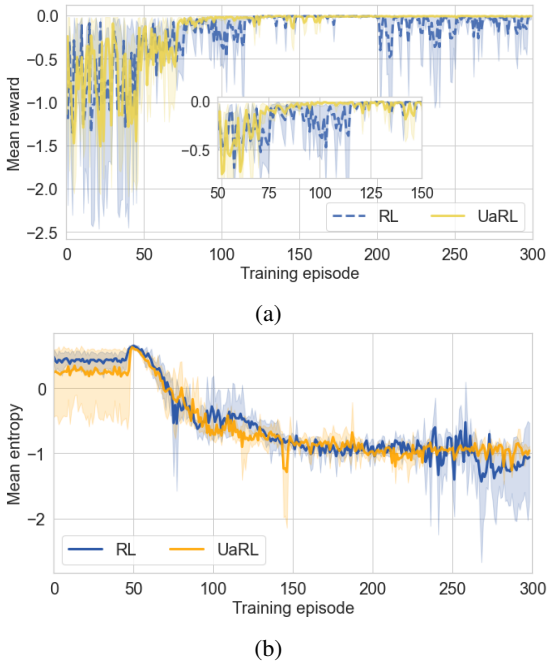


Fig. 4: The (a) reward curve and (b) entropy change of RL-based EMS in the training process.

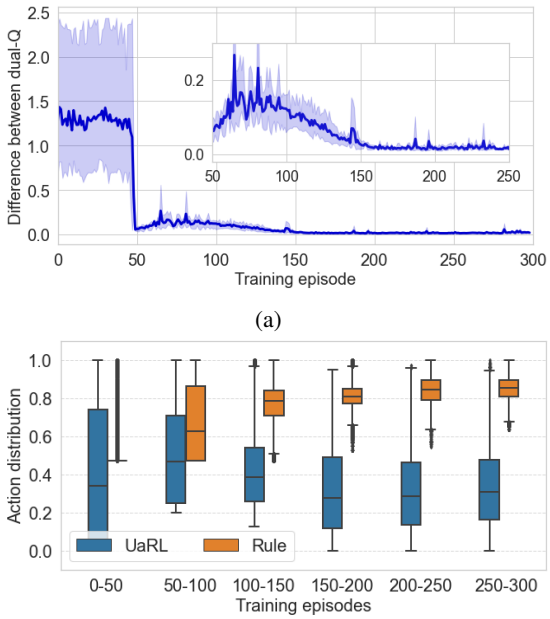


Fig. 5: Training evolution of dual Q-network uncertainty and action distributions. (a) Absolute Q-difference $|Q_1 - Q_2|$ of UaRL over episodes. (b) Action distributions (Normalization) of UaRL and Rule methods at different stages.

A. Simulation Setup

To cover the common driving conditions of various HEVs, we selected NEDC, UDDS, JN1015, LA92, and US06 to construct the training cycles, and randomly selected one cycle for each training episode. To verify the generalization ability of the algorithm, FTP75, HWFET and WLTC which did not appear in the training scenario, were selected as test cycles. In addition, two real-world driving profiles, V01 and V02, were collected and included as supplementary test scenarios [8]. These data sets were obtained from actual vehicle operations and feature diverse speed and acceleration patterns, as illustrated in Table IV and Fig. 3. These cycles represent a wide range of driving conditions, allowing for a comprehensive evaluation of algorithm performance. Specifically, NEDC, FTP75 and WLTC exhibit large speed fluctuations typical of urban driving, while US06 and HWFET reflect high-speed highway conditions. V01 and V02 further extend the evaluation scope by capturing realistic and complex scenarios.

In this work, the initial SoC value in all simulations is set to 0.5. All neural networks are built using the PyTorch toolkit, and both the RL framework and vehicle model are implemented using Python scripts. The simulations are run at a frequency of 1 Hz on a system equipped with a RTX 2060 Max-Q GPU acceleration. To reduce the impact of random fluctuations during RL training, we conducted validation with three distinct random seeds.

To verify the effectiveness of the proposed strategy, we consider the following EMSs as baseline strategies:

- DP [37]: serves as an offline optimal benchmark leveraging full driving cycle knowledge and high computation, offering a theoretical upper bound for performance comparison despite its impracticality for real-time use.
- Rule-based [23]: a heuristic approach that relies on predefined rules to manage energy flow.
- KGRL [7]: A latest knowledge-guided RL EMS that integrates reference strategies derived from expert knowledge as a guiding demonstrator. Improve strategy performance by introducing imitation-aware objectives.
- BEAR [17]: A state-of-the-art offline RL EMS, which introduces an experience replay method that adaptively

selects data based on its value contribution to improve policy robustness and energy efficiency.
- Vanilla RL [13]: this study uses traditional SAC as a baseline, an advanced off-policy, model-free RL approach.

B. Training Performance Evaluation

Fig. 4(a) compares the mean reward of the proposed UaRL method with vanilla RL during training. UaRL converges faster in the early phase (0 to 100 episodes), reaching convergence around 80 episodes, while vanilla RL takes about 120 episodes to achieve similar performance. This shows that UaRL's uncertainty-driven AL mechanism effectively reduces exploration uncertainty and improves policy learning efficiency. In the mid-to-late phase (200 to 300 episodes), UaRL exhibits smaller reward fluctuations, indicating greater stability compared to vanilla RL. Fig. 4(b) shows the mean action entropy during training, indicating uncertainty in action selection. Both methods begin with high entropy due to exploration, which decreases as policies become more deterministic. However, vanilla RL exhibits larger entropy fluctuations later, suggesting less stable learning. In contrast, UaRL maintains lower, more stable entropy, indicating more consistent learning and a smoother transition from exploration to exploitation.

Fig. 5(a) shows the evolution of the Q-network difference ($|Q_1 - Q_2|$) during training. During the initial warm-up phase (first 50 episodes), data is collected without updates, resulting in large, fluctuating Q-differences that reflect high epistemic uncertainty. Once updates begin, the difference rapidly decreases and gradually converges, indicating better model understanding. While heuristic, $|Q_1 - Q_2|$ is a widely accepted proxy for uncertainty, and its reduction typically aligns with policy convergence.

Fig. 5(b) compares the action distributions of UaRL and the Rule method across training stages. The Rule policy depends on real-time SoC, which is indirectly influenced by UaRL actions; hence, its action distribution evolves during training. In the early phase (0–50 episodes), UaRL shows high variance in actions. From episodes 50–100, UaRL's distribution gradually aligns with the Rule-based EMS, highlighting the latter's guiding role. After 100 episodes, UaRL increasingly diverges, showing greater autonomy and adaptability. These results indicate that imitation loss promotes faster early convergence and expert alignment, while long-term learning enables the agent to balance optimality and generalization without strictly imitating the expert.

C. Testing Performance Evaluation

1) Energy consumption assessment: Table V summarizes the test results of different approaches across two training cycles (NEDC, US06), three unseen cycles (HWFET, FTP75 and WLTC), and two real-world cycles (V01, V02). DP Benchmarking (DP Bench.) is total cost ratio relative to DP. Across all driving cycles, the proposed method consistently achieves lower total cost than both other baselines. For example, in the challenging and unseen FTP75 test cycle, the proposed method's total cost is only 4.3% higher than DP. In the US06 training cycle, UaRL even surpasses DP in economic performance, likely because DP focuses on strict SoC maintenance, whereas our method is optimized for overall cost efficiency. While KGRL outperforms vanilla RL by leveraging knowledge guidance, it still lags behind UaRL, particularly in balancing fuel economy and battery health; similarly, BEAR benefits from robust offline training and surpasses both online RL and KGRL in most cases, yet remains less effective than UaRL, especially under high variability. For high-speed cycles (HWFET and US06), where high power demand limits fuel savings, our method demonstrates superior management of battery aging cost compared to other learning-based approaches. In scenarios with frequent speed fluctuations, such as NEDC and FTP75, the proposed method achieves minimal terminal SoC deviation.

Real-world cycles V01 and V02 provide a rigorous generalization test. Vanilla RL shows poor performance and instability, with V02 cost exceeding even the rule-based method. In contrast, our method achieves \$10.45/100 km (V01) and \$10.68/100 km (V02), leading in both cost and SoC control, with low variance across runs. On average, it reduces total cost by 23.71% vs. rule-based, and 14.68% vs. vanilla RL. The method also achieves a DP Benchmarking score of 96.07%, indicating near-optimal performance.

2) Analysis of LIB operation trajectory: Fig. 6 illustrates the LIB operating trajectories under different methods, including SoC and SoH profiles. Notable differences are observed among the approaches. The DP method, as the global optimum, maintains a smooth and stable SoC trajectory, indicating precise energy management, and shows the slowest SoH degradation under most conditions. However, in the high-speed US06 cycle, its SoH degrades faster due to prioritizing SoC maintenance over battery health (see TableV and Fig. 6(b)).

The rule-based method suffers from frequent SoC oscillations, indicating poor control. The RL method shows unstable performance across scenarios. For example, in NEDC and US06, its SoC trajectory is volatile—especially in early and mid NEDC (frequent start-stops) and mid-to-late US06 (sustained high speed)—suggesting inadequate adaptation. In FTP75, RL experiences severe SoC fluctuations, accelerating battery aging. RL tends to overuse the range extender in NEDC and FTP75 (keeping SoC high), but allows SoC to drop too low in high-speed scenarios like US06 and HWFET, revealing policy inefficiencies.

In contrast, UaRL maintains a smoother SoC trajectory, closely matching the DP trend and mitigating sharp fluctuations. In late NEDC, UaRL leverages more electric power to reduce fuel use; in high-speed cycles, it maintains SoC stability, outperforming RL in consistency. In FTP75, UaRL ensures more stable energy use and mitigates energy oscillation. Regarding SoH, UaRL consistently slows battery degradation across all scenarios by optimizing EMS, showing better battery protection than RL.

3) Analysis of engine operating point: The distribution of engine operating points reflects how each method manages engine performance under different conditions. As shown in Fig. 7, the rule-based method concentrates in the high-efficiency region due to its fixed control logic, but this often

TABLE V: Comparison of different methods under various driving cycles.

Cycle	Method	Total cost ($/100 km)	Engine fuel cost ($/100 km)	LIB aging cost ($/100 km)	Fuel consumption (L/100 km)	DP Bench. (%)	Final SoC
NEDC	DP / Rule	11.67 / 15.59	9.27 / 10.07	2.39 / 5.53	9.20 / 9.98	100 / 74.86	0.508 / 0.533
	RL	14.46 (3.01)	10.34 (2.10)	4.12 (0.92)	10.26 (2.10)	80.71	0.540
	KGRL	13.74 (1.11)	9.37 (0.44)	4.37 (1.22)	9.30 (0.44)	84.93	0.512
	BEAR	13.23 (0.82)	9.56 (0.75)	3.67 (0.64)	9.48 (0.76)	88.21	0.507
	Ours	12.34 (0.28)	8.90 (0.38)	3.44 (0.40)	8.72 (0.46)	94.57	0.493
US06	DP / Rule	16.29 / 19.12	12.60 / 13.12	3.69 / 6.00	12.50 / 13.01	100 / 85.20	0.505 / 0.516
	RL	16.37 (1.04)	12.47 (0.48)	3.89 (1.39)	12.30 (0.59)	99.51	0.503
	KGRL	16.07 (0.55)	12.65 (0.37)	3.43 (0.92)	12.55 (0.38)	101.37	0.507
	BEAR	16.12 (0.68)	12.45 (0.23)	3.67 (0.45)	12.34 (0.28)	101.05	0.506
	Ours	15.61 (0.65)	12.42 (0.21)	3.18 (0.61)	12.31 (0.21)	104.36	0.505
FTP75	DP / Rule	11.36 / 16.47	8.09 / 9.10	3.27 / 7.37	8.01 / 9.03	100 / 68.97	0.497 / 0.533
	RL	14.36 (3.07)	9.34 (1.50)	4.73 (0.82)	9.27 (1.49)	79.11	0.559
	KGRL	12.95 (0.86)	8.82 (0.42)	4.12 (1.24)	8.76 (0.41)	87.72	0.514
	BEAR	13.14 (0.55)	8.97 (0.43)	4.17 (0.60)	8.84 (0.42)	86.45	0.508
	Ours	11.85 (0.21)	8.03 (0.02)	3.81 (0.19)	7.93 (0.12)	95.86	0.493
HWFET	DP / Rule	11.65 / 17.17	10.46 / 11.48	1.19 / 5.69	10.38 / 11.39	100 / 67.85	0.503 / 0.532
	RL	13.12 (1.02)	10.85 (0.27)	2.27 (0.81)	10.77 (0.26)	88.80	0.524
	KGRL	12.97 (0.46)	10.71 (0.31)	2.26 (0.48)	10.62 (0.31)	89.82	0.518
	BEAR	12.71 (0.45)	10.55 (0.28)	2.16 (0.18)	10.48 (0.26)	91.66	0.516
	Ours	12.61 (0.39)	10.67 (0.24)	1.94 (0.19)	10.58 (0.25)	92.39	0.515
WLTC	DP / Rule	13.83 / 18.11	11.15 / 12.04	2.69 / 6.07	11.06 / 11.95	100 / 76.37	0.502 / 0.541
	RL	16.84 (1.40)	12.93 (0.65)	3.91 (0.75)	12.75 (0.68)	82.13	0.468
	KGRL	15.56 (0.82)	11.87 (0.53)	3.69 (0.47)	11.77 (0.52)	88.88	0.454
	BEAR	15.41 (0.65)	11.42 (0.42)	3.99 (0.23)	11.33 (0.43)	89.75	0.461
	Ours	14.07 (0.36)	10.19 (0.25)	3.88 (0.21)	9.78 (0.35)	98.29	0.477
V01	DP / Rule	10.06 / 14.49	7.71 / 8.45	2.35 / 6.04	7.65 / 8.39	100 / 69.43	0.502 / 0.540
	RL	13.47 (3.92)	9.31 (2.19)	4.16 (1.73)	9.23 (2.16)	74.68	0.570
	KGRL	12.68 (0.63)	8.69 (0.52)	3.99 (0.48)	8.62 (0.49)	79.34	0.520
	BEAR	12.86 (0.57)	8.84 (0.40)	4.02 (0.33)	8.75 (0.42)	78.23	0.515
	Ours	10.45 (0.08)	7.64 (0.03)	2.81 (0.10)	7.58 (0.04)	96.27	0.506
V02	DP / Rule	9.93 / 13.45	7.19 / 7.39	2.74 / 6.06	7.11 / 7.31	100 / 73.83	0.495 / 0.496
	RL	13.87 (4.14)	9.10 (2.76)	4.77 (1.37)	9.01 (2.77)	71.59	0.627
	KGRL	11.22 (0.79)	7.55 (0.65)	3.67 (0.58)	7.49 (0.60)	88.50	0.513
	BEAR	11.14 (0.64)	7.33 (0.37)	3.81 (0.43)	7.20 (0.35)	89.14	0.496
	Ours	10.68 (0.46)	7.06 (0.22)	3.62 (0.25)	6.95 (0.26)	92.98	0.491

Note: The energy consumption metrics is represented as the "mean (standard deviation)" under different random seeds.

sacrifices flexibility, SOC stability, and battery health. In contrast, the DP method, as a global optimum, shows a broader distribution, allowing flexible trajectory planning and better adaptation to varying conditions. RL-based methods yield similar patterns, while UaRL achieves a wider spread with higher average efficiency, indicating a better balance between adaptability and performance. UaRL supports robust multi-peak strategies while maintaining efficient engine operation.

Fig. 8 presents the 3D distributions of engine operating points under the US06 and HWFET cycles. The DP method consistently shows more frequent and widespread clusters in the high-efficiency region. RL-based methods generally follow DP's trend while retaining some exploration. Among them, UaRL achieves more frequent and broader coverage of high-efficiency zones, highlighting its enhanced adaptability and control diversity under dynamic environments. This confirms that effective engine management is not merely about maximizing operation in the peak-efficiency region, but about ensuring frequent efficient operation within a broader range. Such distribution supports better real-world robustness and overall system performance.

D. Ablation Study

To evaluate the impact of each core component in the proposed UaRL framework, ablation studies were conducted as shown in Fig. 9. Specifically, the A_1 term was removed from Eq. (34), the A_2 term from Eq. (38), and both terms together to assess their combined effect. The imitation loss term ω_I was also excluded from Eq. (35), and its initial weight was varied between 1 and 0.1 to examine sensitivity. In addition, the sliding window length M and active query threshold th were adjusted to evaluate their influence on AL triggering.

Results under the FTP75 cycle show that the complete model ("Ours") achieves the lowest cost, validating the effectiveness of all components in improving energy efficiency and battery health. Removing A_1 or A_2 increases fuel consumption and LIB usage, with both removed causing the largest performance drop. Excluding ω_I leads to higher fuel use, indicating its benefit for fuel efficiency though with limited impact on battery usage. Notably, setting ω_I to 1 throughout training

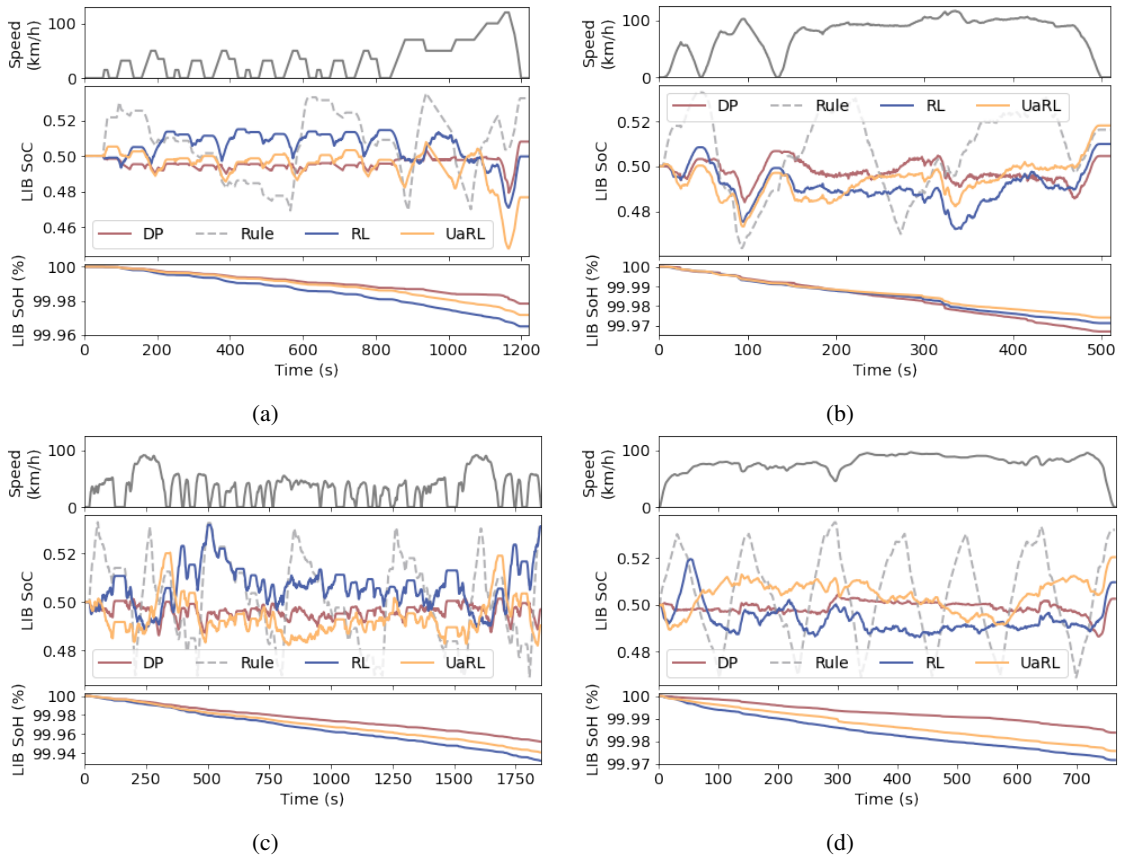


Fig. 6: The operation trajectories of Li-ion battery under different EMS are presented for the following driving cycles: (a) NEDC, (b) US06, (c) FTP75, and (d) HWFET.

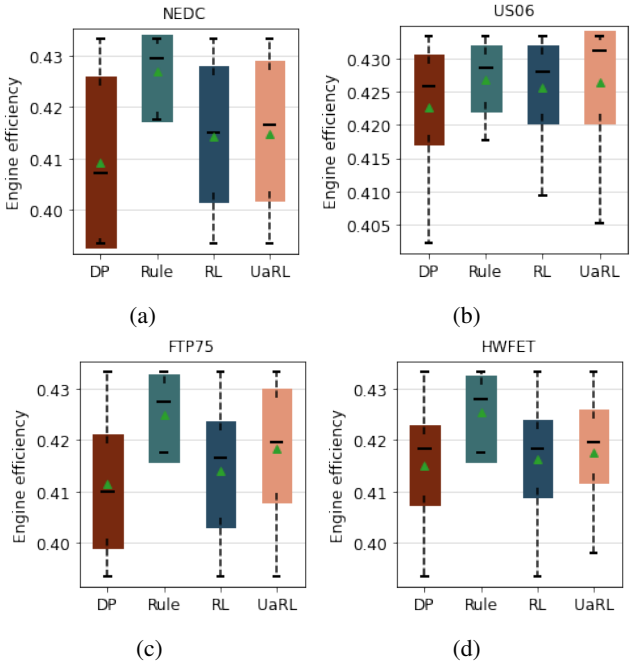


Fig. 7: The distribution of engine operating points across different test driving cycles is shown for: (a) NEDC, (b) US06, (c) FTP75, and (d) HWFET.

TABLE VI: Computational complexity evaluation of EMS.

Strategy	Clock time elapse per step (s)
Ours	1.38e-3
RL-based	1.21e-3
Rule-based	4.24e-4

significantly degrades performance, suggesting that excessive imitation of rule-based strategies can be harmful. In contrast, initializing ω_I at 0.1 only causes slight degradation, supporting the value of moderate imitation. Finally, performance remains stable across variations in M and th, indicating robustness to these hyperparameters.

E. Calculation Efficiency Analysis

To evaluate real-time feasibility, we compared the average computation time per step across different EMS strategies on a laptop equipped with an RTX 2060 Max-Q GPU, as shown in Table VI. The results show that our UaRL method incurs only slightly more computational overhead than the standard RL-based approach and remains close to that of the rule-based baseline. Although all strategies complete decision-making within a few milliseconds on the test platform, further deployment-level optimization may be required to meet the resource constraints of embedded automotive controllers.

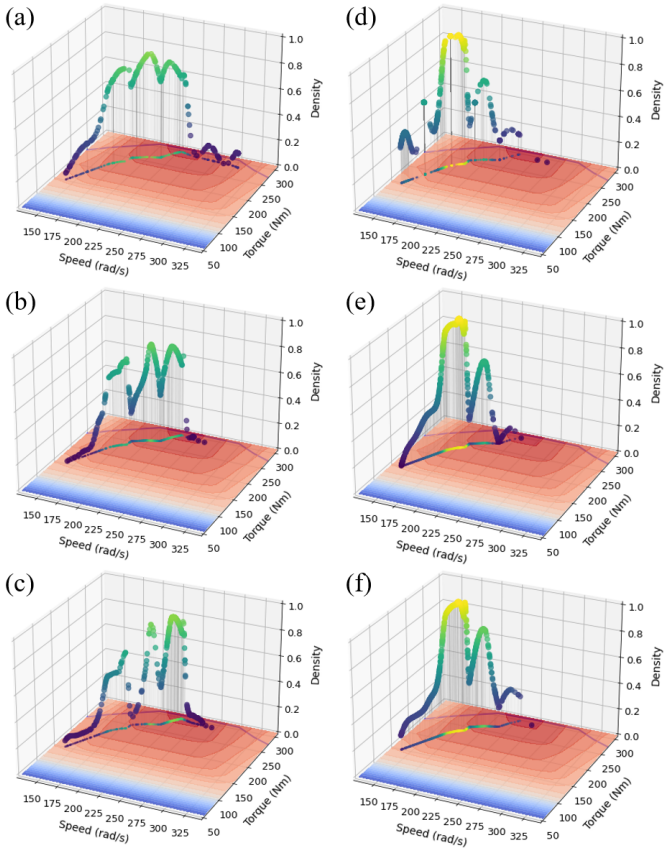


Fig. 8: 3D distributions of engine operating points under US06 ((a) DP, (b) RL, (c) UaRL) and HWFET ((d) DP, (e) RL, (f) UaRL) driving cycles.

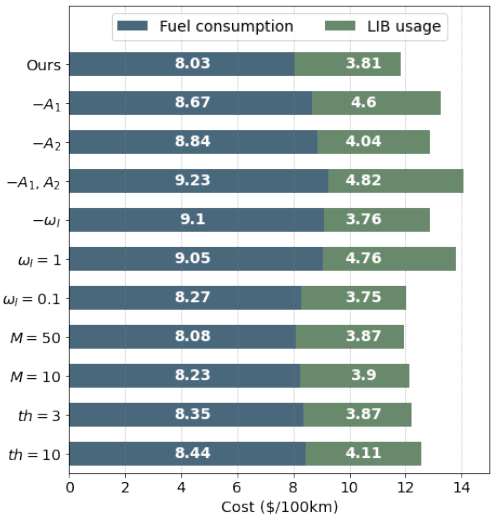


Fig. 9: Ablation test under FTP75 cycle.

VI. Summary and Conclusions

This paper presents an uncertainty-aware active RL framework for EMS, aiming to endow it with self-awareness and active learning capabilities. The specific conclusion is as follows:

Faster convergence: UaRL achieves faster convergence during the early training phase compared to vanilla RL, reducing exploration costs and improving training efficiency.

Superior economic performance: Compared with vanilla RL, UaRL improves the economic performance of the studied medium-sized REEB by 14.68%, achieving near-global optimality and strong generalization in unseen testing scenarios.

Enhanced battery management: UaRL demonstrates excellent SoC stability, effectively avoiding large fluctuations seen in baselines, while achieving slower battery SoH degradation, leading to better long-term battery health.

Engine optimization: UaRL improves engine operation efficiency by balancing dynamic adaptability, achieving robust energy management in diverse driving conditions.

While UaRL generalizes well in standard cycles, real-world applications may present more extreme conditions. Future work will focus on enabling continual learning during deployment to help models adapt to changing vehicle environments. To ensure efficient deployment of lightweight and updated models, we will explore model pruning, quantization, and vehicle-cloud collaboration for online policy updates. Furthermore, we plan to investigate the use of more flexible expert sources to replace rule-based controller, such as human demonstrations or large model-based feedback, to improve the diversity and effectiveness of advice.

References

[1] I. Aghabali, J. Bauman, P. J. Kollmeyer, Y. Wang, B. Bilgin, and A. Emadi, "800-v electric vehicle powertrains: Review and analysis of benefits, challenges, and future trends," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 3, pp. 927–948, 2021.

[2] I. E. Agency, "Co2 emissions in 2023," International Energy Agency Report, 2024. [Online]. Available: https://www.iea.org/reports/co2-emissions-in-2023

[3] ——, "Global ev outlook 2024," International Energy Agency Report, 2024. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2024

[4] Y. Li, H. He, Y. Chen, and H. Wang, "A cloud-based eco-driving solution for autonomous hybrid electric bus rapid transit in cooperative vehicle-infrastructure systems: A dynamic programming approach," *Green Energy and Intelligent Transportation*, vol. 2, no. 6, p. 100122, 2023.

[5] H. He, X. Meng, Y. Wang, A. Khajepour, X. An, R. Wang, and F. Sun, "Deep reinforcement learning based energy management strategies for electrified vehicles: Recent advances and perspectives," *Renewable and Sustainable Energy Reviews*, vol. 192, p. 114248, 2024.

[6] D. Hu, C. Huang, J. Zhao, Y. Zhao, and J. Wu, "Autonomous driving economic car-following motion strategy based on adaptive rollout model-based policy optimization," *IEEE Transactions on Transportation Electrification*, 2025.

[7] X. Li, H. He, and J. Wu, "Knowledge-guided deep reinforcement learning for multiobjective energy management of fuel cell electric vehicles," *IEEE Transactions on Transportation Electrification*, vol. 11, no. 1, pp. 2344–2355, 2024.

[8] Y. Wang, H. He, Y. Wu, P. Wang, H. Wang, R. Lian, J. Wu, Q. Li, X. Meng, Y. Tang, *et al.*, "Learningems: A unified framework and open-source benchmark for learning-based energy management of electric vehicles," *Engineering*, 2024.

[9] B. Luo, Z. Wu, F. Zhou, and B.-C. Wang, "Human-in-the-loop reinforcement learning in continuous-action space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 15 735–15 744, 2023.

[10] D. Hu, H. Xie, K. Song, Y. Zhang, and L. Yan, "An apprenticeship-reinforcement learning scheme based on expert demonstrations for energy management strategy of hybrid electric vehicles," *Applied Energy*, vol. 342, p. 121227, 2023.

[11] S. Chaudhari, P. Aggarwal, V. Murahari, T. Rajpurohit, A. Kalyan, K. Narasimhan, A. Deshpande, and B. Castro da Silva, "Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms," *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–72, 2024.
[12] D. Hu, C. Huang, J. Wu, and X. Yuan, "Toward multi-task generalization in autonomous navigation: A human-in-the-loop adversarial reinforcement learning with diffusion policy," *IEEE Transactions on Intelligent Transportation Systems*, 2025.
[13] D. Hu, C. Huang, J. Wu, H. Wei, and D. Pi, "Enhancing data-driven energy management strategy via digital expert guidance for electrified vehicles," *Applied Energy*, vol. 381, p. 125138, 2025.
[14] T. A. Berrueta, A. Pinosky, and T. D. Murphey, "Maximum diffusion reinforcement learning," *Nature Machine Intelligence*, pp. 1–11, 2024.
[15] S. D. Ghode and M. Digalwar, "Deep dyna reinforcement learning based energy management system for solar operated hybrid electric vehicle using load scheduling technique," *Journal of Energy Storage*, vol. 102, p. 114106, 2024.
[16] D. Hu, C. Huang, G. Yin, Y. Li, Y. Huang, H. Huang, J. Wu, W. Li, and H. Xie, "A transfer-based reinforcement learning collaborative energy management strategy for extended-range electric buses with cabin temperature comfort consideration," *Energy*, vol. 290, p. 130097, 2024.
[17] Z. Niu, J. Wu, and H. He, "A novel experience replay-based offline deep reinforcement learning for energy management of hybrid electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 72, no. 7, pp. 7160–7169, 2025.
[18] J. Wu, J. Peng, M. Li, and Y. Wu, "Enhancing fuel cell electric vehicle efficiency with tip-ems: A trainable integrated predictive energy management approach," *Energy Conversion and Management*, vol. 310, p. 118499, 2024.
[19] J. Wu, Z. Wei, H. He, H. Wei, S. Li, and F. Gao, "Ensembled traffic-aware transformer-based predictive energy management for electrified vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 12 333–12 346, 2024.
[20] M. Li, X. Wan, M. Yan, J. Wu, and H. He, "Attentive hybrid reinforcement learning-based eco-driving strategy for connected vehicles with hybrid action spaces and surrounding vehicles attention," *Energy Conversion and Management*, vol. 321, p. 119059, 2024.
[21] C. Jia, H. He, J. Zhou, J. Li, Z. Wei, K. Li, and M. Li, "A novel deep reinforcement learning-based predictive energy management for fuel cell buses integrating speed and passenger prediction," *International Journal of Hydrogen Energy*, vol. 100, pp. 456–465, 2025.
[22] W. Chen, J. Peng, Y. Ma, H. He, T. Ren, and C. Wang, "Eco-driving framework for hybrid electric vehicles in multi-lane scenarios by using deep reinforcement learning methods," *Green Energy and Intelligent Transportation*, p. 100309, 2025.
[23] J. Wu, C. Huang, H. He, and H. Huang, "Confidence-aware reinforcement learning for energy management of electrified vehicles," *Renewable and Sustainable Energy Reviews*, vol. 191, p. 114154, 2024.
[24] J. Wu, J. Chen, and D. Huang, "Entropy-based active learning for object detection with progressive diversity constraint," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9397–9406.
[25] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, "Scalable active learning for object detection," in *2020 IEEE intelligent vehicles symposium (iv)*. IEEE, 2020, pp. 1430–1435.
[26] S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, "Advanced active learning strategies for object detection," in *2020 IEEE intelligent vehicles symposium (IV)*. IEEE, 2020, pp. 871–876.
[27] S. Hanneke and L. Yang, "Toward a general theory of online selective sampling: Trading off mistakes and queries," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3997–4005.
[28] S. Schulze and O. Evans, "Active reinforcement learning with monte-carlo tree search," *arXiv preprint arXiv:1803.04926*, 2018.
[29] D. Krueger, J. Leike, O. Evans, and J. Salvatier, "Active reinforcement learning: Observing rewards at a cost," *arXiv preprint arXiv:2011.06709*, 2020.
[30] A. D. Tucker, C. Biddulph, C. Wang, and T. Joachims, "Bandits with costly reward observations," in *Uncertainty in Artificial Intelligence*. PMLR, 2023, pp. 2147–2156.
[31] C. Lork, W.-T. Li, Y. Qin, Y. Zhou, C. Yuen, W. Tushar, and T. K. Saha, "An uncertainty-aware deep reinforcement learning framework for residential air conditioning energy management," *Applied Energy*, vol. 276, p. 115426, 2020.
[32] S. Onori, L. Serrao, and G. Rizzoni, *Hybrid electric vehicles: Energy management strategies*. Springer, 2016, vol. 13.
[33] Z. Wei, Z. Quan, J. Wu, Y. Li, J. Pou, and H. Zhong, "Deep deterministic policy gradient-drl enabled multiphysics-constrained fast charging of lithium-ion battery," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2588–2598, 2022.
[34] S. Ebbesen, P. Elbert, and L. Guzzella, "Battery state-of-health perceptive energy management for hybrid electric vehicles," *IEEE Transactions on Vehicular technology*, vol. 61, no. 7, pp. 2893–2900, 2012.
[35] Z. P. Cano, D. Banham, S. Ye, A. Hintennach, J. Lu, M. Fowler, and Z. Chen, "Batteries and fuel cells for emerging electric vehicle markets," *Nature energy*, vol. 3, no. 4, pp. 279–289, 2018.
[36] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.
[37] L. Deng, S. Li, X. Tang, K. Yang, and X. Lin, "Battery thermal-and cabin comfort-aware collaborative energy management for plug-in fuel cell electric vehicles based on the soft actor-critic algorithm," *Energy Conversion and Management*, vol. 283, p. 116889, 2023.

Dong Hu (Graduate Student Member, IEEE) received the B.Eng. degree from Wuhan University of Technology, Wuhan, China, in 2020, and the M.Eng. degree from Tianjin University, Tianjin, China, in 2023. He is currently pursuing the Ph.D. degree with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong. His current research interests include reinforcement learning, autonomous vehicle navigation, and electrified vehicle optimization.

Cheng Tian (Graduate Student Member, IEEE) received the M.Eng. degree in vehicle engineering from Tongji University, Shanghai, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR, China. His research interests include vehicle dynamics and control, safe planning, and control for autonomous vehicles.

Anh-Tu Nguyen (Senior Member, IEEE) received the Ph.D. degree in automatic control from the University of Valenciennes, Valenciennes, France, in 2013. He is currently an Associate Professor with INSA Hauts-de-France, Université Polytechnique Hauts-de-France, Valenciennes. His research interests include robust control and estimation, cybernetics control systems, and human–machine shared control.

Chao Huang (Senior Member, IEEE) is a Research Assistant Professor at the Department of Industrial and System Engineering, The Hong Kong Polytechnic University. She received her Ph.D. degree from the University of Wollongong in 2018. Her research interests are human-machine collaboration, mobile robots, and path planning. She serves as an Associate Editor for IEEE Transactions on Transportation Electrification, Engineering Applications of Artificial Intelligence, Control Engineering Practice, etc.