Ecological Driving Framework of Hybrid Electric Vehicle Based on Heterogeneous Multi-Agent Deep Reinforcement Learning

Jiankun Peng, Weiqi Chen, Yi Fan, Hongwen He, *Senior Member, IEEE*, Zhongbao Wei, *Senior Member, IEEE*, and Chunye Ma

***Abstract*— Hybrid electric vehicles (HEVs) have great potential to be discovered in terms of energy saving and emission reduction, and ecological driving provides theoretical guidance for giving full play to their advantages in real traffic scenarios. In order to implement an ecological driving strategy with the lowest cost throughout the life cycle in a car-following scenario, the safety and comfort, fuel economy, and battery health need to be considered, which is a complex nonlinear and multiobjective coupled optimization task. Therefore, a novel multi-agent deep deterministic policy gradient (MADDPG) based framework with two heterogeneous agents to optimize adaptive cruise control (ACC) and energy management strategy (EMS), respectively, is proposed, thereby decoupling optimization objectives of different domains. Because of the asynchronous of multi-agents, different learning rate schedules are analyzed to coordinate the learning process to optimize training results; an improvement on the prioritized experience replay (PER) technique is proposed, which improves the optimization performance of the original MADDPG method by more than 10%. Simulations under mixed driving cycles show that, on the premise of ensuring car-following performance, the overall driving cost, including fuel consumption and battery health degradation of the MADDPG-based method, can reach 93.88% of that of DP, and the proposed algorithm has good adaptability to different driving conditions.**

***Index Terms*— Adaptive cruise control (ACC), battery health, deep deterministic policy gradient, ecological driving, energy management, heterogeneous multi-agent.**

NOMENCLATURE

P_{eng}	Power of diesel engine, kW.
T_{eng}	Torque of diesel engine, Nm.
W_{eng}	Rational speed of engine, rpm.
η_{eng}	Efficiency of engine fuel.
P_{gen}	Power of generator, kW.
T_{gen}	Torque of generator, Nm.
W_{gen}	Rational speed of generator, rpm.
η_{gen}	Efficiency of generator.
P_{batt}	Charge/discharge of battery, kW.
C_n	Nominal capacity of battery, Ah.
SoC	State of charge.
SoH	State of health.

Manuscript received 25 December 2022; revised 19 February 2023 and 21 April 2023; accepted 17 May 2023. Date of publication 22 May 2023; date of current version 16 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 52072074, in part by the Fundamental Research Funds for the Central Universities under Grant 2242021R40007, and in part by the "Zhishan" Scholars Programs of Southeast University. *(Corresponding authors: Jiankun Peng; Hongwen He.)*

Jiankun Peng, Weiqi Chen, Yi Fan, and Chunye Ma are with the School of Transportation, Southeast University, Nanjing 211102, China (e-mail: jkpeng@seu.edu.cn; 220213486@seu.edu.cn; yffan024@gmail.com; cma@seu.edu.cn).

Hongwen He and Zhongbao Wei are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: hwhebit@bit.edu.cn; weizb@bit.edu.cn).

Digital Object Identifier 10.1109/TTE.2023.3278350

I. INTRODUCTION

AS FOSSIL fuel crisis and environmental pollution continue to intensify with the increase in car ownership, the transportation sector urgently needs to explore effective solutions to save energy and reduce emissions. Vehicle technology and vehicle usage are two main factors affecting vehicle emissions and fuel consumption [1]. Hybrid electric vehicles (HEVs) with dual energy sources of internal combustion engines (ICE) and power battery pack have become one of the preferred solutions to achieve energy saving and emission reduction in the transportation section [2].

Ecological driving (eco-driving) refers to an optimal control problem (OCP) that can minimize energy consumption from the perspective of vehicle driving [3]. Particularly, eco-driving can enable road vehicles to realize better fuel economy when considering the influences of the surrounding traffic environment [4]. As for driving the economy, eco-driving also needs to take the health performance of the power battery into consideration due the high price and underlying degradation of the power battery [5]. Thus, for HEVs, an extensive ecological driving concept is to save fuel and keep battery health as much as possible during driving.

Fuel consumption and health performance are deeply related to vehicle dynamics and powertrain operation (VD&PT) [3], while VD&PT are determined by driving characteristics such as velocity and acceleration. The deep coupling between driving characteristics and economic performance makes eco-driving a complex multiobjective optimization problem. It should be noted that driving behavior is mainly determined by traffic environment and driver personality, which is weakly related to the vehicle itself. While fuel consumption and health performance have a deep correlation with VD&PT. From this point of view, eco-driving tasks can be decomposed into two subproblems. One is optimization of fuel consumption and battery health, which is a microscopic OCP inside vehicle.

And the other is control of vehicle driving behavior in a traffic environment, which focuses on macroscopic traffic situations. The former can be formulated as energy management strategy (EMS), and the latter can be different tasks because of specific scenarios [6], such as approach and departure in a signal intersection, free cruising up/downhill, and car-following. The two subproblems seem to be independent but actually interact with VD&PT.

A. EMS for HEVs

The essential idea of eco-driving is to reduce fuel consumption and battery degradation during driving. EMS plays a crucial role in affecting the performance, reliability, fuel economy, and battery health of HEVs by distributing power among different energy storages [7]. Current EMS can be classified into three types: rule-based, optimization-based, and learning-based. A typical rule-based EMS can reduce fuel consumption per 100 km from 25.46 L diesel to 22.80 L diesel [8]; however, the high requirement of expertise and unsatisfactory optimization performance hinder their further application [9], and the preset rules also limit the flexibility under different driving cycles [10]. Optimization-based methods such as dynamic programming (DP) [11], equivalent cost minimization strategy (ECMS) [12], Pontryagin's minimum principle (PMP) [13], and model predictive control (MPC) [14], [15], have been proved to be effective to achieve near-optimal fuel consumption with lesser dependence on intuition and experience of professional engineers [16]. The power battery pack onboard HEV is not only expensive but also has a shorter life cycle, whose health state is another important optimization object of EMS besides fuel consumption [17], and PMP-based EMS considering battery health are reported in [18] and [19]. DP and PMP are both computationally intensive and not conducive to real-time application in their usual forms. The biggest problem of ECMS-based EMS is that the sensitive equivalence factor can only be tuned appropriately when the driving cycle is known a priori, which is usually not feasible in real-world conditions [12]. MPC method requires a linear approximation of nonlinear VD&PT model, which weakens the accuracy of optimal control [3].

Deep reinforcement learning (DRL) methods make up for the above shortcomings. EMS is modeled as the Markov decision process (MDP) [20], and decision logic between state variables and control actions is implicitly constructed through deep neural networks (DNN), thereby preserving the control accuracy of nonlinear VD&PT models [21]. The decision network is iteratively trained using a large amount of data collected in various interactive environments, which not only can achieve near-optimal multiobjective optimization performance of the DP benchmark [22], but also the diversity of data ensures the applicability of the control strategy in different scenarios [23]. More importantly, the policy parameters are trained in cloud servers and deployed to the onboard control unit (OBU) through vehicle-to-infrastructure (V2I) communication, ensuring real-time application capability [24]. Recently, battery health has been considered in DRL-based EMS [25], [26]. In summary, existing EMS researches based on traffic-free environment provide a solid theoretical basis for the development and validation of eco-driving strategy based on traffic scenarios.

B. Eco-Driving in Different Scenarios

As stated before, eco-driving can be implemented in three main driving scenarios [6]: 1) approach and departure in a signal intersection, 2) free cruising up/downhill, 3) and car-following.

Wu et al. [26] a proposed prediction-based ecological approach-departure strategy applied to urban intersections, saving 1.9% of energy and reducing standard pollutant emissions by 1.9%–33.4%. Gao et al. [27] proposed an optimization-based generation of reference velocity, the so-called "eco-driving cycle," to reduce energy consumption for battery electric vehicles when cruising on mountain roads. Bai et al. [28] reduced energy consumption by 12.70% and saved 11.75% travel time by hybrid reinforcement learning-based eco-driving strategy at signalized intersections.

Compared with urban intersections and free cruising scenes, the car-following scenario has more energy-saving potential for implementing eco-driving [29]. A pulse-and-glide (PnG) cruise strategy improved fuel economy by up to 20% in automated car-following is presented [30]. As an automated vehicle control technology, adaptive cruise control (ACC) has received widespread attention through eco-driving in a car-following scenarios [31]. Li et al. [32] proposed an ecological ACC for parallel HEV in car-following scenario to improve fuel economy and maintain a desired intervehicle distance based on heuristic DP. Chada et al. [33] implemented an MPC based ecological ACC to perform robust vehicle following and PnG to optimally control the engine ON and OFF states and save additional fuel. Vajedi et al. [34] developed a framework to calculate the globally optimal solution of ACC through PMP and control vehicle speed through nonlinear MPC technology, which reduces the total energy cost by 19% in a car-following scenario.

C. Motivation and Contribution

Under the complete eco-driving concept for HEV, there are two subproblems: 1) energy management considering battery health inside the vehicle powertrain system; 2) vehicle control in traffic scene, where ACC in a car-following situation is focused in this article. How to optimize the two tasks collaboratively is our original intention.

Control methods of eco-driving in existing researches are mainly two types: rule-based [30] and optimization-based [32], [33], [34]. Rule-based methods draw support on daily driving experience (such as slow acceleration) to design velocity control algorithms. Although they are practical, their applicability and optimization effects are not satisfactory due to changeable traffic environments. Optimization-based methods incorporate traffic information into hierarchical control architecture, and optimized speed profiles bring higher energy efficiency. The limited amount of information interaction and the independent optimization framework limit, however, algorithm performance, which is difficult to achieve multiobjective

collaborative optimization and online application. In view of the above deficiencies, the DRL algorithm is applied to the collaborative optimization of ACC and EMS for the first time in [35], which makes a contribution to the development of learning-based eco-driving methods.

ACC is a kinematic control problem in the transportation domain, while EMS deals with power distribution inside HEV. Both are multiobjective optimization problems but have different dimensions in time and space scales. So they need to be moderately decoupled, otherwise facing difficult trade-off problems [35]. Hierarchical control architecture can coordinate weights between the ACC and EMS through the decoupling of upper and lower layers but the optimization-based algorithm cannot be applied online and need to be improved to have better performance [36]. Since ACC and EMS belong to different fields and have Markov properties, they can be optimized by integrated multi-agent deep reinforcement learning (MADRL) algorithm. With the centralized training and decentralized execution architecture of multi-agent deep deterministic policy gradient (MADDPG), it not only decouples the two subproblems of eco-driving to avoid weight imbalance affecting performance, but also retains the online application advantage of integrated architecture.

Current researches on MADDPG as a control algorithm focuses on fields of UAV clusters [37], connected autonomous vehicles [38], communication equipment [39], and dynamics analysis of energy saving in intelligent buildings [40], [41]. Their control objects are physically homogeneous, which means that each agent has the same inputs, outputs, and targets. ACC and EMS, however, cannot be constructed as two homogeneous agents due to the essential differences in optimization tasks, which is a challenge in the application for algorithms. In this work, we propose a novel eco-driving framework for a series HEV (SHEV) based on MADDPG algorithm. The main contributions are as follows:

1) Eco-driving for HEV is decomposed into two subproblems in an extensive framework: one is EMS, and the other is ACC in a car-following scenario. The MADDPG algorithm is employed to collaboratively optimize and synchronously control the eco-driving problem for the first time, where the two heterogeneous agents have both cooperative and competitive relationships.

2) The two agents are fully optimized during cooperation and competition. Agent ACC provides Agent EMS with velocity and acceleration curves with lower overall energy consumption, and Agent EMS can specify a safe and comfortable power allocation strategy for Agent ACC.

3) Different learning rate schedules, including fixed and cyclical are explored to coordinate heterogeneous multi-agent learning synchronization for optimal solutions, and an improvement on the prioritized experience replay (PER) technique is implemented for better performance.

4) The proposed method is not sensitive to initial states, and has good adaptability to different driving cycles such as city, suburban, and highway. This demonstrates its excellent application ability.

The remainder of this article is organized as follows. In Section II, the models of subtasks of eco-driving strategy are described separately. In Section III, main ideas of the MADDPG algorithm and specific application methods are depicted. In Section IV, tests are designed to evaluate the proposed approach, and simulation results are analyzed. The last section concludes this article.

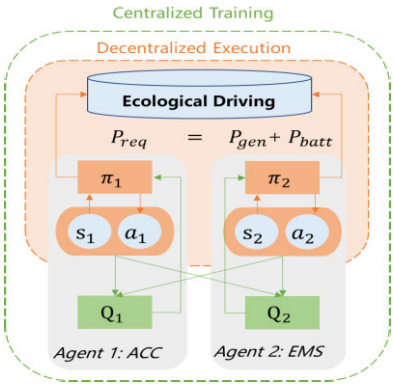

Fig. 1. Relationship between multiple agents.

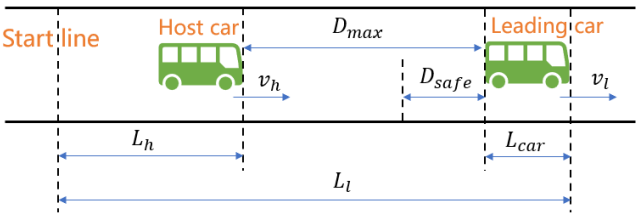

Fig. 2. Car following scenario.

II. System Model Description

This article proposes a MADDPG framework containing two heterogeneous agents, named Agent ACC and Agent EMS, respectively, as shown in Fig. 1. The Agent ACC corresponds to ACC in a car-following scenario, while Agent EMS handles energy management inside a vehicle. Every agent has a centralized value function Q_i and a decentralized policy function π_i. The value functions consider global states and actions, while the policy functions only concern their own states and execute their own actions.

A. ACC Model

The proposed eco-driving framework considers a car-following scenario, where the host car can detect the velocity and acceleration of the leading car and spacing between the two vehicles at each time step by laser and ultrasonic radar. The speed of the leading vehicle follows standard driving cycles, which are predefined. Fig. 2 shows the car-following scenario, where v, a, L are velocity, acceleration, and traveling mileage, respectively, and subscript h, l represents the host car and the leading car, respectively. $D_{h,l}$ is distance between the two vehicles. The car length L_{car} is 5 m. The acceleration of host car a_h is outputted by *Agent ACC*. The velocity and distance can be calculated as follows. In order to ensure the safety and effectiveness of car-following, the speed and

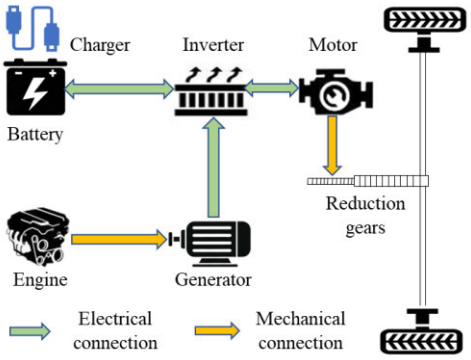

Fig. 3. Structure of the SHEV configuration.

acceleration ranges are set, which are both larger than that of standard driving cycles.

$$\begin{cases} v = \int a dt \\ L = \int v dt \\ D_{h,l} = L_l - L_h - L_{car} \\ 0 \le v_h \le 33.33 \text{ m / s} \\ -2.5 \text{ m / s}^2 \le a_h \le 2.5 \text{ m/s}^2. \end{cases} \tag{1}$$

By controlling the acceleration of the host car, the *Agent ACC* keeps a safe and appropriate distance during the following while paying attention to ride comfort. The maximum distance D_{max} [42] and safe distance D_{safe} [43] are both calculated according to $v_h(t)$. The safe distance D_{safe} is seen as the minimal value of $D_{h,l}$.

$$\begin{cases} D_{max}(t) = 0.0825 * v_h(t)^2 + v_h(t) + 10 \\ D_{safe}(t) = v_h(t) * t_d + (v_h(t))^2 / a_{max} + d_0. \end{cases} \tag{2}$$

Here: t_d is the sum of braking delay and reaction time whose value is 1.5 s [44], d_0 is the safety distance from the leading car after the host car stopped and is 3 m, a_{max} is the maximum acceleration under emergency and equals 6.68 m/s^2.

B. Powertrain Model of SHEV

The vehicle studied is a mid-size passenger van, the main parameters of which are shown in Table I. As shown in Fig. 3, the energy of electric traction motors comes from two parts, which are engine-generator set (EGS) and Li-ion battery (LIB) pack. Therefore, the powertrain of SHEV has two submodels; one is the EGS model, and the other is the LIB model. The next two sections give detailed descriptions, respectively.

C. EGS Model

Since this article focuses on the distribution mechanism of energy between EGS and LIB pack, the driving force from motors is considered to be evenly distributed between two axles. The propelled power requested by motors, which is primarily decided by the curb weight, velocity, and acceleration of vehicle is crucial for the energy management system, the main function of which is to allocate power. Given the

TABLE I
MAIN PARAMETERS OF THE SHEV SPECIFICATION

Symbol	Parameter	Value
P_{eng}^{max}	Peak power of the diesel engine	62 kW
T_{eng}^{max}	Peak torque of the diesel engine	227 Nm
W_{eng}^{max}	Peak speed of the diesel engine	3500 rpm
T_{gen}^{max}	Peak torque of the generator	277 Nm
W_{gen}^{max}	Peak speed of the generator	4000 rpm
T_{mot}^{max}	Peak torque of the traction motor	320 Nm
W_{mot}^{max}	Peak speed of the traction motor	7200 rpm
m	mass of the SHEV	3500 kg
A_w	Windward area	3.9 m^2
R_{wh}	Wheel radius	0.447 m
L_{wh}	wheelbase	2.65 m
i_0	Main reducer ratio	5.857
C_n	Nominal capacity of battery pack	7.42 kWh

acceleration and velocity of the vehicle, the total power requested P_{req} is as follows:

$$\begin{cases} P_{req} = v \cdot F_{req} \\ F_{req} = F_a + F_r + F_i + F_w \\ F_a = m \cdot a \\ F_r = \mu m g cos\theta \\ F_i = m g sin\theta \\ F_w = A_w C_d v^2 / 21.15 \end{cases} \tag{3}$$

F_{req} is the total traction required by car, which is also the total resistance during driving. There are four main components of resistance: inertial force F_a, rolling resistance F_r, resistance due to road slope F_i and aerodynamic drag F_w. The rolling resistance coefficient is denoted by μ, assuming 0.01. Air drag coefficient is denoted by C_d, which is 0.65. θ is road slope, which is considered 0 in this study, and g is the acceleration of gravity, 9.8 m/s^2.

Assuming that the EGS can respond quickly when receiving control signals, the quasi-static fuel and power consumption models are established from efficiency maps in Fig. 4. Torque and speed balance equation describes the transfer between the engine and generator

$$T_{eng} = T_{gen}, \quad W_{eng} = W_{gen}. \tag{4}$$

According to current torque and speed, the efficiency of the engine fuel and generator can be obtained through the efficiency maps, respectively, and then the output power value can be calculated as

$$\begin{cases} P_{eng} = T_{eng} \cdot W_{eng} \\ P_{gen} = T_{gen} \cdot W_{gen} \cdot \eta_{gen}. \end{cases} \tag{5}$$

Given the gasoline lower heating value denoted by $G(4.25 \times 10^7$ J/kg), the fuel consumption rate of engine is

$$\dot{m_f} = \frac{P_{eng}}{G \cdot \eta_{eng}}. \tag{6}$$

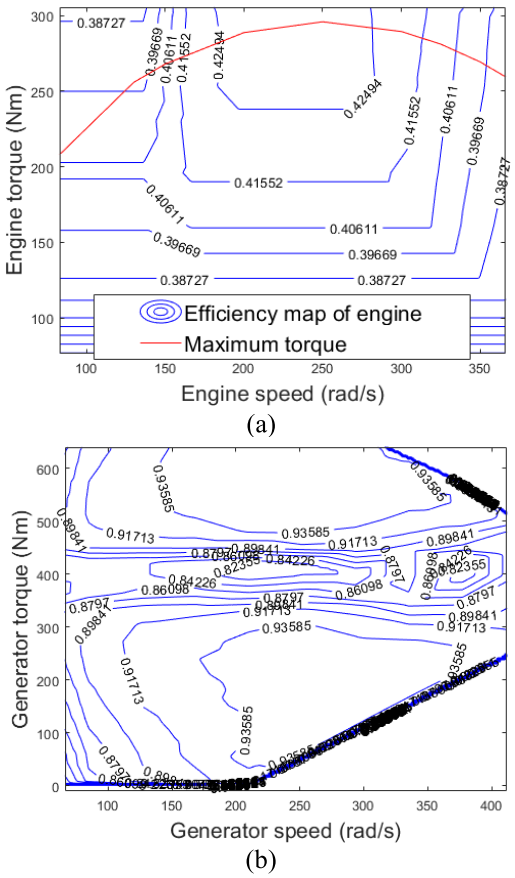

Fig. 4. Efficiency maps. (a) Engine. (b) Generator.

Meanwhile, the boundary constraints of torque and speed must be satisfied for both the engine and generator.

$$\begin{cases} T_{eng}^{min} \le T_{eng} \le T_{eng}^{max} T_{gen}^{min} \le T_{gen} \le T_{gen}^{max} \\ W_{eng}^{min} \le W_{eng} \le W_{eng}^{max} W_{gen}^{min} \le W_{gen} \le W_{gen}^{max}. \end{cases} \tag{7}$$

The power requested by electric traction comes from generator and battery pack, as follows, where η_{inv} denotes the efficiency of the inverter, assuming that regenerative braking is fully adopted as

$$P_{req} = \left(P_{batt} + P_{gen}\right) \cdot \eta_{inv}. \tag{8}$$

D. LIB Model

The LIB is simulated by an electrothermal-aging model, which comprises three subclasses i.e., a second-order *RC* electro model, a two-state thermal model, and an energy-throughput aging model [25]. As shown in Fig. 5, the electro and thermal model is coupled to predict the electrothermal dynamics of LIB. The voltage source in the electro model describes the open-circuit voltage, which is dependent on SoC, while R_s is the total equivalent ohmic resistance. There are some polarization effects inside LIB packs when they are working, such as charge transfer, diffusion phenomena, and passivation layer effects on electrodes [45]. The two *RC* branches are used to simulate the above circumstance. The

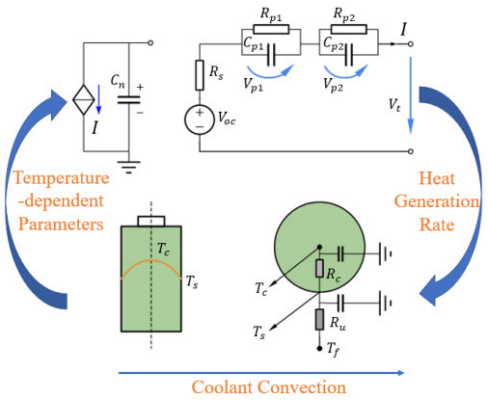

Fig. 5. Coupled electro-thermal model.

governing equations of electro submodel are given by [46]

$$\frac{dSoC(t)}{dt} = \frac{I(t)}{3600C_n} \tag{9}$$

$$\frac{dV_{p1}(t)}{dt} = -\frac{V_{p1}(t)}{R_{p1}(t)C_{p1}(t)} + \frac{I(t)}{C_{p1}(t)} \tag{10}$$

$$\frac{dV_{p2}(t)}{dt} = -\frac{V_{p2}(t)}{R_{p2}(t)C_{p2}(t)} + \frac{I(t)}{C_{p2}(t)} \tag{11}$$

$$V_t(t) = V_{oc}(SoC) + V_{p1}(t) + V_{p2}(t) + R_s I(t) \tag{12}$$

where $I(t)$ and $V_t(t)$ are the load current and terminal voltage at time step t, V_{p1} and V_{p2} are the polarization voltage across the *RC* branches, which are parameterized by the capacitance C_{p1}, C_{p2} and resistance R_{p1}, R_{p2}. According to the thermal energy conservation principle, the following equation is given

$$C_c \frac{dT_c(t)}{dt} = \frac{T_s(t) - T_c(t)}{R_c} + H(t) \tag{13}$$

$$C_s \frac{dT_s(t)}{dt} = \frac{T_c(t) - T_s(t)}{R_c} + \frac{T_f(t) - T_s(t)}{R_u} \tag{14}$$

$$T_a(t) = \frac{T_c(t) + T_s(t)}{2} \tag{15}$$

where T_s, T_c, T_a, and $T_f(t)$ are temperature of battery surface, core, internal average and ambient, respectively, all in the unit of °C. R_c and R_u are the thermal resistances caused by the heat conduction inside the battery and the convection at battery surface. C_c and C_s are equivalent thermal capacitances of the battery core and surface. Ohmic heat, polarization heat and irreversible entropy heat together affect the heat generation rate, which is represented by $H(t)$ inside LIB. The heat generation rate can be calculated by the following equation, where E_n denotes the entropy change during electrochemical reactions.

$$H(t) = I(t)\left[V_{p1}(t) + V_{p2}(t) + R_s(t)I(t)\right] + I(t)[T_a(t) + 273]E_n(SoC, t). \tag{16}$$

Lin et al. [47] determined electrical parameters through a series of pulse relaxation tests for A123 26 650 LIB, the thermal parameters were then identified by the heat generation rate based on the measured current and voltage of the electric model. The established electrothermal model is highly accurate, limiting the root mean square error (RMSE) of the

TABLE II
DEPENDENCE OF PRE-EXPONENTIAL FACTOR TO C-RATE

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

terminal voltage to within 20 mV, while the RMSE of the core and surface temperatures are both below 1 °C during typical driving cycles [26].

Ebbesen et al. [18] developed the energy-throughput model for evaluating battery degradation in their study. It assumes that the LIB can withstand a certain amount of accumulated charge flow before it is scrapped. The dynamic of SoH is hence given by

$$\frac{dSoH(t)}{dt} = -\frac{\int_0^t |I(\tau)|d\tau}{2N(c,T_a)C_n} \tag{17}$$

where $N(c,T_a)$ is the equivalent number of cycles till the LIB reaches its end of life (EOL). In order to calculate the instantaneous change of SoH, (17) is rewritten in discrete-time form

$$\Delta SoH_t = -\frac{|I(t)|\Delta t}{2N(c,T_a)C_n} \tag{18}$$

where Δt is the current duration. The Arrhenius equation-based empirical model of capacity loss takes into account the impact of C-rate (c) and internal temperature, the equation is as follows:

$$\Delta C_n = B(c)\cdot \exp\left(-\frac{E_a(c)}{RT_a}\right)\cdot Ah^z \tag{19}$$

where ΔC_n is the percentage of capacity loss, $B(c)$ denotes preexponential factor referred to Table II, R is the ideal gas constant, which equals to 8.314 J/(mol · K), z is the power-law factor equals to 0.55, Ah represents the ampere-hour throughput, and E_a denotes the activation energy in the unit of J/mol defined by [18]

$$E_a(c) = 31700 - 370.3\cdot c. \tag{20}$$

The LIB reaches its EOL when the C_n drops by 20%. According to this definition and referring to [18], Ah and N can be derived as

$$Ah(c,T_a) = \left[\frac{20}{B(c)}\cdot \exp\left(-\frac{E_a(c)}{RT_a}\right)\right]^{1/z} \tag{21}$$

$$N(c,T_a) = 3600\cdot Ah(c,T_a)/C_n. \tag{22}$$

Finally, the change in SoH for a given current, temperature, and service dynamics can be calculated according to (18) to understand the aging condition of the battery pack.

III. MADDPG FOR ECO-DRIVING

This work is devoted to finding a comprehensive eco-driving strategy involving ACC and EMS. This leads to two basic optimization tasks: 1) to achieve safe and comfortable car-following performance; and 2) to achieve EMS with the lowest overall driving cost, which consists of fuel consumption and battery health degradation. The proposed heterogeneous multi-agent framework actually solves a multiobjective optimization problem where each agent handles an optimization task of a different domain. Fig. 6 shows the overall architecture of the proposed framework.

A. Problem Formulation

The most essential idea of *RL* is that agent obtains reward $r(t)$ of the executed action $a(t)$ under the state $s(t)$ during interaction with environment, and learns an optimal strategy π by maximizing the expectation of cumulative discount reward $R = E[\sum_{t=0}^{T}\gamma^t r_t]$. Where T and γ are total time steps and discount factor, respectively. The goal of the proposed method is to minimize the various cost K incurred by vehicles during driving, so that the reward function of agent can be constructed

$$r_i(t) = -K_i(t) \tag{23}$$

where t is time step, the subscript i indicates different agents, number 1 for *Agent ACC* and number 2 for *Agent EMS*.

1) Agent ACC: The purposes of *Agent ACC* is to maintain a safe distance between the leading and following cars, meanwhile keeping a comfortable acceleration. *Agent ACC* achieves these goals by minimizing the costs $K_1(t)$ as follows:

$$K_1(t) = \omega_1 C_s(t) + \omega_2 C_c(t) \tag{24}$$

where ω_1 and ω_2 represent weights of different objectives, and they are determined in our previous work [35]. $C_s(t)$ is safe cost, and $C_c(t)$ is the comfort cost. Safety is the highest priority when driving and the distance is used to measure it

$$C_s(t) = \begin{cases} v_h^{max}, & D_{h,l}(t) \le 0 \\ v_h, & 0 < D_{h,l}(t) \le D_{safe}(t) \\ D_{h,l}(t) - D_{max}(t), & D_{h,l}(t) > D_{max}(t) \\ 0, & otherwise. \end{cases} \tag{25}$$

The host car should be severely punished when it collides with the leading car in simulation environment; thus, the maximum speed is considered a safety cost. When the distance is less than the safety distance, the velocity of the host car is seen as the safety cost, which means the slower the speed, the lesser the cost. When the following distance is greater than the maximum following distance, the difference between the two is regarded as safety cost. In addition, *Agent ACC* controls the change rate of acceleration (*jerk*) to ensure ride comfort

$$\begin{cases} jerk(t) = a_h(t) - a_h(t-1) \\ a_r = \max(a_h) - \min(a_h) \\ C_c(t) = |jerk(t)|/a_r \end{cases} \tag{26}$$

where a_r denotes the instantaneous variation range of the host vehicle acceleration, which equals to 5 in this study.

2) Agent EMS: The *Agent EMS* interacts with the powertrain and battery system of SHEV in the form of energy flow. To achieve EMS with the lowest driving cost, there are three objectives: 1) reduce fuel consumption, 2) keep charge within a reasonable range, and 3) decrease LIB health degradation. Correspondingly, its cost $K_2(t)$ comes from three parts, the

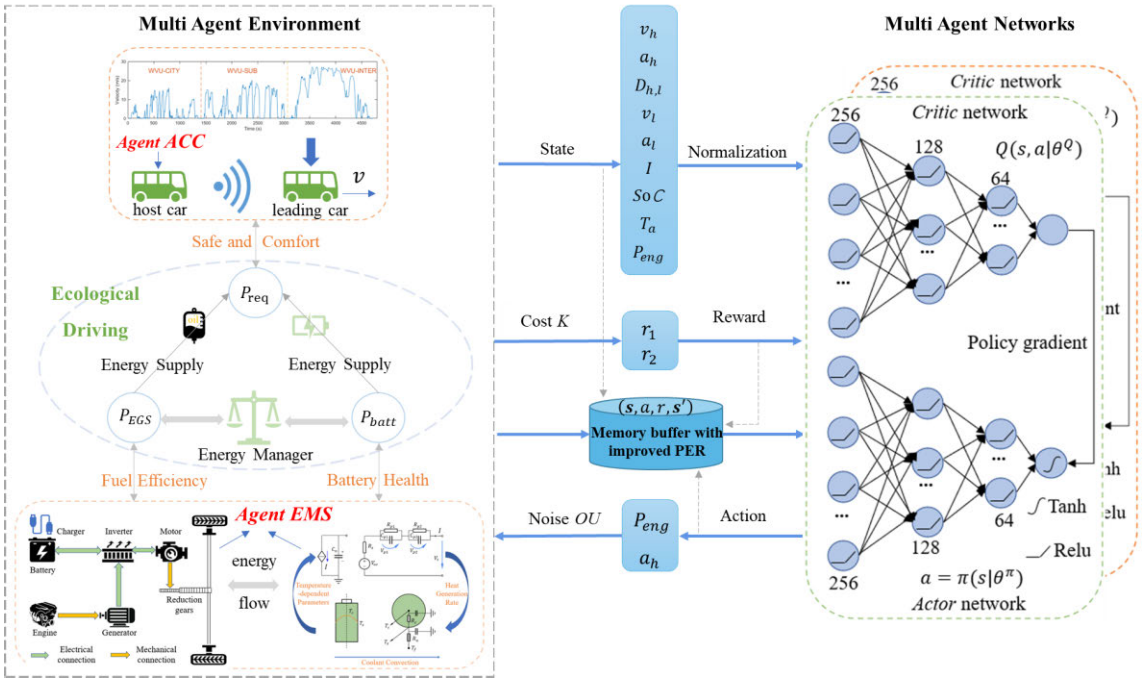

Fig. 6. Overall architecture of the MADDPG based framework.

fuel consumption $\dot{m_f}(t)$, the SoC depletion $C_{soc}(t)$ and battery wear $C_{soh}(t)$.

$$\begin{cases} K_2(t) = \omega_3 \dot{m_f}(t) + \omega_4 C_{soc}(t) + \omega_5 C_{soh,t} \\ C_{soc}(t) = |SoC(t) - SoC_{tar}| \\ C_{soh}(t) = \Delta SoH_t \end{cases} \tag{27}$$

where ω_3 denotes the money expense on 1 kg fuel, ω_4 and ω_5 are transition coefficients enforcing the SoC mismatch and capacity loss dimensionally compatible with the fuel consumption rate. ω_5 is defined as a ratio of the battery replacement cost to the cost of 1 kg of gasoline [19]. And SoC_{tar} is the reference value of SoC.

B. Multi-Agent Deep Deterministic Policy Gradient

DDPG is a typical single-agent DRL algorithm of *Actor-Critic* architecture that deals with continuous state and action problems [48]. In order to break the correlation between training data to ensure learning performance, DDPG draws on the successful experience of DQN and uses the experience replay technique. The MADDPG algorithm is an extended version of DDPG, considering an environment with multiple agents, where each agent has an *Actor* network, a *Critic* network, a target *Actor* network, and a target *Critic* network [49]. The parameters of target networks are softly updated to stabilize the training process. The basis of the MADDPG algorithm is centralized training with decentralized execution, which means each agent is associated with an *Actor* network only taking its own observation and a centralized *Critic* network taking strategies of all other agents, as shown in Fig. 1. The agent feeds *Actor* network with state vector observed by itself to get control actions to execute, and then obtains corresponding rewards and new state from the interactive environment. Thus, the *Actor* network is updated by the state-action value $Q_\pi(\boldsymbol{s}, a)$. It is worth noting that the *Actor* network outputs actions based on the local observation during execution.

The joint policy of all agents is denoted by $\boldsymbol{\pi} = [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_n]$, where policy $\boldsymbol{\pi}_i$ is actually a deep neural network parameterized by θ_i, then derives policy parameter set $\theta = [\theta_1, \theta_2, \ldots, \theta_n]$. The expectation of cumulative discount reward of agent i is

$$J(\theta_i) = E_{s\sim p^\pi, a_i\sim\pi_i}\left[\sum_{t=0}^{T} \gamma^t r_t^i\right] \tag{28}$$

where p^π is the state distribution, θ_i, which is implicit in $\boldsymbol{\pi}_i$ represents the probability distribution function of action.

The policy gradient is calculated as follows [47]:

$$\nabla_{\theta_i} J(\theta_i) = E_{\boldsymbol{s},a\sim M}\left[\nabla_{\theta_i}\pi_i(a_i|o_i)\nabla_{a_i} Q_i^\pi(\boldsymbol{s}, a_1, \ldots, a_n)|_{a_i=\pi_i(o_i)}\right] \tag{29}$$

where M is experience buffer consisting of series of tuples $(\boldsymbol{s}, \boldsymbol{s}', a_1, \ldots, a_n, r_1, \ldots, r_n)$, which records what agents have experienced. o_i is the observation of agent i, $\boldsymbol{s} = [o_1, \ldots, o_n]$ is the state space, which consists of the observations of all agents and $\boldsymbol{s}' = [o_1', \ldots, o_n']$ denotes the next observations of all agents when they have executed the actions. While $Q_i^{\boldsymbol{\pi}}(\boldsymbol{s}, a_1, \ldots, a_n)$ is the centralized state-action value function of agent i. This algorithm realizes competition and cooperation among multi-agent environments under the premise that each agent has a different state-action value function [48]. The centralized state-action value function $Q_i^{\boldsymbol{\pi}}$ is updated as

$$\begin{cases} L(\theta_i) = E_{\boldsymbol{s},a,r,s'}\left[\left(Q_i^\pi(\boldsymbol{s}, a_1, \ldots, a_n) - y_i\right)^2\right] \\ y_i = r_i + \gamma Q_i^{\pi'}(\boldsymbol{s}', a_1', \ldots, a_n')|_{a_j'=\pi_j'(o_j')} \end{cases} \tag{30}$$

where $\delta_i = Q_i^{\pi}(\boldsymbol{s}, a_1, \ldots, a_n) - y_i$ is called "*TD-error*," and $\boldsymbol{\pi}' = [\boldsymbol{\pi}'_1, \ldots, \boldsymbol{\pi}'_n]$ is the set of target policy networks with delayed parameters θ'_i. Since the actions of all agents are known for each agent, for any $\boldsymbol{\pi}_i \neq \boldsymbol{\pi}'_i$, there is

$$\begin{aligned} &P\left(\boldsymbol{s}' \mid \boldsymbol{s}, a_1, \ldots, a_n, \boldsymbol{\pi_1}, \ldots, \boldsymbol{\pi_n}\right) = P\left(\boldsymbol{s}' | \boldsymbol{s}, a_1, \ldots, a_n\right) \\ &= P\left(\boldsymbol{s}' \mid \boldsymbol{s}, a_1, \ldots, a_n, \boldsymbol{\pi'_1}, \ldots, \boldsymbol{\pi'_n}\right). \end{aligned} \tag{31}$$

Thus, the environment becomes stable when real-time policies change. Afterward, the parameters of the target *Actor* network and target *Critic* network are softly updated, where τ is the soft factor to control updating magnitude.

$$\begin{cases} \theta_i'^{\boldsymbol{\pi}} \leftarrow \tau\theta_i^{\boldsymbol{\pi}} + (1-\tau)\theta_i'^{\boldsymbol{\pi}} \\ \theta_i'^{Q} \leftarrow \tau\theta_i^{Q} + (1-\tau)\theta_i'^{Q}. \end{cases} \tag{32}$$

C. Improvement of PER Technique

Experience replay technique is an essential part of off-policy *RL* algorithm, which saves the data collected by anagent during training and reuses them randomly for multiple times. PER [52] is an effective variant, which develops a method to replay important transitions more frequently to improve learning efficiency. In DDPG, the probability of sampling transition i is defined as

$$\begin{cases} P(i) = \dfrac{p_i^{\alpha}}{\sum_k p_k^{\alpha}} \\ p_i^{\alpha} = |\delta_i| + \epsilon = |Q_i^{\pi}(s,a) - y_i| + \epsilon \end{cases} \tag{33}$$

where p_i^{α} is the priority of transition i and calculated by *TD-error*. There are two agents in our proposed method, which means two different *TD-errors* to update the priority for twice during one training episode. This can lead to instability of the experience pool negative impact on algorithm performance, which is discussed detailly in Section IV-B. To avoid it, we improved (33) by using the average of two agents' *TD-errors*to update p_i^{α} just once during one training episode

$$p_i^{\alpha} = \frac{1}{2}(|\delta_1| + |\delta_2|) + \epsilon \tag{34}$$

where α determines how much prioritization is used, ϵ is a small positive constant to prevent zero priority. In this work, $\alpha = 0.4, \epsilon = 1e-6$.

D. Training of MADDPG-Based Eco-Driving Strategy

MADRL-based algorithm can be formulated by defining state space, action space, and reward function, while the state and action vectors of each agent are as follows.

1) State Space: The state space contains what the agent observed from the simulation environment, it is the input to the agent's *Actor* network and determines what action the agent should take. Hence, state space ought to be composed of key information for making decisions.

$$\begin{cases} \boldsymbol{s} = [\boldsymbol{s_1}, \boldsymbol{s_2}] \\ \boldsymbol{s_1} = \left[v_h, a_h, D_{h,l}, v_l, a_l\right] \\ \boldsymbol{s_2} = \left[I, SoC, T_a, P_{req}\right] \end{cases} \tag{35}$$

where $\boldsymbol{s_1}, \boldsymbol{s_2}$ denotes the observation of *Agent ACC* and *Agent EMS*, respectively. According to previous work experience, the state vectors are initialized as: $\boldsymbol{s_1} = [0, 0, 15, 0, 0]$, $\boldsymbol{s_2} = [0, 0.6, 25, 0]$. Given that the vehicle is static at the first few seconds, so $v_h, a_h, v_l, a_l, I, P_{req}$ can be all initialized to 0. And battery temperature T_a is the same as the environment before driving, 25 °C. In order to avoid overfitting, initial values of $D_{h,l}$ and SoC are randomly selected in adaptability experiments in Section IV-D.

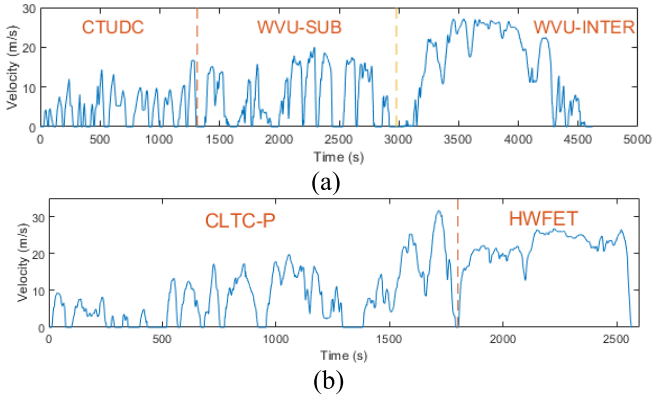

Fig. 7. Driving cycles for training and validation. (a) Mix-train, trip for training. (b) Mix-valid, trip for validating.

2) Action Space: For better convergence and optimization effect, action space should not only consider the problems to be optimized but also pay attention to the structure of the reward function. When people drive, they actually control the acceleration or deceleration of the car by changing the position of the accelerator or brake pedal, thereby realizing speed control; therefore, the acceleration is chosen as the action of *Agent ACC*.

The energy demand of SHEV is satisfied by ICE and LIB, and (9) describes this relationship where P_{req} is determined by *Agent ACC*. Once one of the P_{batt} and P_{gen} is controlled by *RL* agent, the other can be derived from (9). Note that P_{gen} can be derived by (5) and (6) directly from P_{eng}. From the efficiency maps in Fig. 4, the agent can easily choose highly efficient operation points, so that the engine power P_{eng} is selected as the action of *Agent EMS*.

$$\begin{cases} \boldsymbol{a} = [a_1, a_2] \\ a_1 = a_h \\ a_2 = P_{eng}. \end{cases} \tag{36}$$

Different from previous *RL* algorithms, DDPG needs to add random processes to action to realize exploration, so as to learn various possible strategies. MADDPG inherits this feature, and the Ornstein-Uhlenbeck process is used in this article

$$a_i(t) = \pi_i(\boldsymbol{s}_i(t)) + OU\left(0, \sigma_t^2\right) \tag{37}$$

where σ_t denotes the standard deviation of random noise and decays continuously over training to balance exploration and exploitation. The initial value of σ_t is 0.25, and it decays exponentially every episode with a decay rate of 0.999.

3) Datasets: As shown in Fig. 7, to ensure that the trained agents can adapt to different driving conditions, a mixture cycle (*Mix-train*) of low to medium speed and highspeed conditions is constructed as a training dataset, which includes city, suburban, and highway conditions. And mixture cycle

TABLE III
CHARACTERISTICS OF SELECTED DRIVING CYCLES

Driving Cycle	Max. Vel. (m/s)	Avg. Vel. (m/s)	Max. Accel. (m/s^2)	Max. Decel. (m/s^2)	$\frac{Time_{acc}}{Time_{dec}}$	Trip time (s)	Trip mileage (km)
CTUDC	16.67	4.49	0.91	-1.04	1.44	1314	5.898
WVU-SUB	20.02	7.19	1.29	-2.16	1.28	1665	11.969
WVU-INTER	27.15	15.22	1.42	-1.86	1.06	1640	24.958
Mix-train	27.15	9.27	1.42	-2.16	1.26	4619	42.825
CLTC-P	31.67	8.04	1.92	-1.94	1.11	1800	14.48
HWFET	26.77	21.54	1.43	-1.48	1.14	766	16.503
Mix-valid	31.67	12.07	1.92	-1.94	1.12	2566	30.983

(*Mix-valid*) of China's light-duty vehicle test cycle-passenger car (CLTC-P) and highway fuel economy test cycle (HWFET) is constructed as the trip for testing the proposed algorithm to evaluate its robustness. The characteristics of each driving cycle are summarized in Table III, which indicates that the test trip is statistically different from the training trip, making relevant verifications more objective. Key hyperparameters are as listed in Table IV, and Algorithm 1 shows the pseudocode of the proposed MADDPG-based eco-driving strategy.

Algorithm 1 MADDPG-Based Eco-Driving Strategy

Initialize *Actor* and *Critic* networks for two agents;
Initialize *OU* noise and experience replay buffer M;
for episode = 1 to 500 **do**:
  Reset environment and observe initial states $\boldsymbol{s}(0)$;
  for time step $t = 1$ to trip time **do**:
    For each variable x_j in $\boldsymbol{s}$: store x_j into X_j;
    For agent i: choose action $\boldsymbol{a_i} = \pi_{\theta^i}(\boldsymbol{s_i}) + OU_t$;
    Execute actions $\boldsymbol{a} = [a_1, a_2]$;
    Obtain reward $\boldsymbol{r} = [r_1, r_2]$;
    Observe next state $\boldsymbol{s'} = [\boldsymbol{s_1}', \boldsymbol{s_2}']$;
    Store $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{s'})$ into replay buffer M;
    Update state matrix: $\boldsymbol{s} \leftarrow \boldsymbol{s'}$;
    if replay buffer M is full, **do**:
      Sample minibatch of N samples $(\boldsymbol{s}^k, \boldsymbol{a}^k, \boldsymbol{r}^k, \boldsymbol{s}'^k)$ with probability $P(k) = (p_k^\alpha / \sum_m p_m^\alpha)$
      for agent $i = 1$ to 2 **do**
        Compute importance-sampling weight:

$$W_i = \frac{1}{N^\beta \cdot P(k)^\beta \cdot max_i w_i}$$

        Set $y_i^k = r_i + \gamma Q_i^{\boldsymbol{\pi}'}(\boldsymbol{s}'^k, a_1', a_2')|_{a_j' = \boldsymbol{\pi}_j'(o_j')}$
        Compute *TD-error*$\delta_i = Q_i^{\boldsymbol{\pi}}(\boldsymbol{s}^k, a_1^k, a_2^k) - y_i^k$
        Update *Critic* network by minimizing the loss:

$$L(\theta_i) = \frac{1}{N} \sum_k W_i \cdot {\delta_i}^2$$

        Updated *Actor* using sampled policy gradient:

$$\nabla_{\theta_i} J = \frac{1}{N} \sum_k \nabla_{\theta_i} \boldsymbol{\pi}_i(a_i|o_i) \nabla_{a_i} Q_i^{\boldsymbol{\pi}}(\boldsymbol{s}^k, a_1^k, a_2^k)|_{a_i = \boldsymbol{\pi}_i(o_i)}$$

      end for
      Update the priority of transition k:

$$p_k^\alpha = \frac{1}{2}(|\delta_1| + |\delta_2|) + \epsilon$$

      Update *Actor* and *Critic* target network of agent i :

$$\theta_i'^Q \leftarrow \tau \theta_i^Q + (1-\tau)\theta_i'^Q$$
$$\theta_i'^{\boldsymbol{\pi}} \leftarrow \tau \theta_i^{\boldsymbol{\pi}} + (1-\tau)\theta_i'^{\boldsymbol{\pi}}$$

    end if
  end for
end for
Output final parameterized policy network $\boldsymbol{\pi}$.

IV. RESULTS AND DISCUSSIONS

In this chapter, the influences of learning rate on optimization results are first studied, and the optimal learning rate schedule is determined. Then the improvement of PER in MADDPG is discussed, and finally the optimality and adaptability of the proposed method are analyzed. The established driving cycle *Mix-train* is used to train parameters of neural networks, and the *Mix-valid* cycle is used to evaluate the adaptability of the proposed method.

A. Learning Rate Schedules

From the point of view of game theory, the training process of the MADRL algorithm is essentially a game process of multiple agents to achieve Nash equilibrium [54]. Since the critic network of each agent in MADDPG includes states and actions of other agents, which means that the global information is obtained, the strategy of each agent will affect other agents' behaviors. In ideal situation, with appropriate hyperparameter tuning, each agent converges to an optimal policy synchronously. Among numerous hyperparameters, the learning rate has the most significant impact on the optimization results of the DDPG algorithm, and MADDPG inherits this feature. The rest of this section shows the optimization performance corresponding to different learning rate schedules.

The results are described in terms of reward value of agents, battery health, fuel consumption, and SoC trajectory. Table V presents learning rate schedules to be compared, where triangular#1 and triangular#2 represent different learning rate cycles, as shown in Fig. 8. They are called cyclical learning rates [50], and the process of cyclically changing the learning rate between reasonable boundary values eliminates the time-consuming experimentation to find global optimal learning rate. The exploration of learning rate schedules in this article will have reference significance for applications of MADDPG in other engineering fields. In addition, the step size is set to 50, which is one-tenth of the total training episodes.

Fig. 9 depicts the optimization results when executing eight different learning rate schedules, respectively. It should be pointed out that other parameters of these experiments are exactly the same. Judging from the average reward values obtained by agents in each training episode, when schedule#8

TABLE IV
KEY HYPERPARAMETERS OF THE MADDPG ALGORITHM

Parameter Description	Value
Hidden layer number of *Actor* and *Critic* (target) networks	3
Neurons distribution of *Actor* and *Critic* (target) networks	256, 128, 64
Neural networks connection	Fully connected
Optimizer	Adam
Learning rate scheduler	Cyclical [53]
Learning rate range of *Actor of Agent ACC*	[1e-5, 1e-3]
Learning rate range of *Critic of Agent ACC*	[1e-4, 5e-3]
Learning rate range of *Actor of Agent EMS*	[1e-5, 1e-4]
Learning rate range of *Critic of Agent EMS*	[1e-5, 5e-4]
Discounting factor	0.975
Noise discount rate	0.999
Size of experience replay buffer	1e5
Minibatch size	64
Number of training episodes	500

TABLE V
LEARNING RATE SCHEDULES

schedule	agent Driver		agent Energy		cyclic way
	actor LR	critic LR	actor LR	critic LR	
#1	1e-4	1e-3	1e-5	1e-4	fixed + fixed
#2	[1e-5,1e-3]	[1e-4,1e-2]	1e-5	1e-4	triangular#1+fixed
#3	[1e-5,1e-3]	[1e-5,1e-2]	1e-5	1e-4	triangular#2+fixed
#4	[1e-5,1e-3]	[1e-4,1e-2]	[1e-6,1e-4]	[1e-5,1e-3]	triangular#1+triangular#1
#5	[1e-5,1e-3]	[1e-5,1e-2]	[1e-6,1e-4]	[1e-6,1e-3]	triangular#2+triangular#1
#6	[1e-5,1e-3]	[1e-4,1e-2]	[1e-5,1e-4]	[1e-4,1e-3]	triangular#2+triangular#2
#7	[1e-5,1e-3]	[1e-4,1e-2]	[1e-5,5e-4]	[1e-4,1e-3]	triangular#2+triangular#2
#8	[1e-5,1e-3]	[1e-4,5e-3]	[1e-5,5e-4]	[1e-5,5e-4]	triangular#2+triangular#2

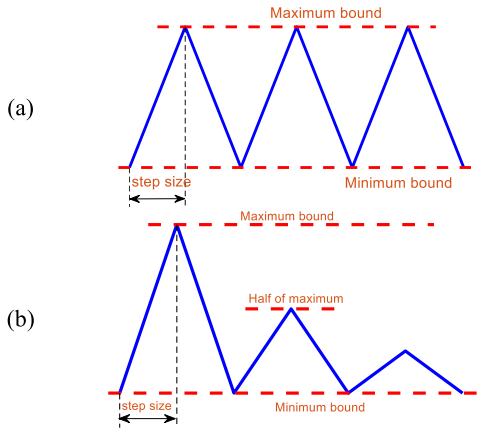

Fig. 8. Two kinds of learning rate cycle: (a) triangular #1 and (b) triangular #2.

is used, the *Agent ACC* gets a higher and more stable reward curve than other schedules in the later stage of training [Fig. 9(a)]; the *Agent EMS* achieves the smoothest and most stable, even though schedule#3 has slightly higher reward values [Fig. 9(b)]. The above two curves also prove the convergence of the proposed algorithm.

In terms of ACC in a car-following scenario, from Fig. 9(c) we can see that all the configurations can achieve safe and effective car following except schedule#2 and schedule#3. And schedule#8 has a shorter following distance in low-speed sections, which means more efficient usage of road resources. More importantly, as shown in Fig. 9(d), the variance of the absolute value of the acceleration of schedule#8 is the smallest among all tests, which represents the most comfortable car following control.

From the perspective of optimization performance of EMS, the schedule#8 maintains battery health to the greatest extent [Fig. 9(f)], meanwhile, learns an excellent fuel-saving strategy, which shows the best fuel economy [Fig. 9(e)] and a reasonable SoC trajectory [Fig. 9(g)]. In the complete driving cycle, compared with other schedules, the SoC trajectory corresponding to schedule#8 changes smoothly and always remains in a larger and more reasonable range, which confirms once again that the optimal EMS has been learned.

In summary, schedule#8 is finally selected as the learning rate scheduler adopted by the proposed eco-driving algorithm due to its excellent performance in both ACC and EMS.

TABLE VI
COMPARISON OF FUEL COST AND OVERALL DRIVING COST

	#MADDPG	*#MADDPG -PER*	*#MADDPG-improved-PER*
Fuel cost (L/100km)	8.24	9.13	7.38
Comparison	100%	110.80%	89.56%
Overall driving cost	CNY84.90	CNY100.71	CNY77.75
Comparison	100%	118.62%	91.58%

B. Experience Replay Technique

This section aims to evaluate performance of the improved PER technique. *#MADDPG* is the basic version of our proposed method, which is uniform sampling. *#MADDPG-PER* is the variant using (33), and *#MADDPG-improved-PER* is the variant using the improved (34). As presented in Fig. 10(a), *#MADDPG-PER* performs worse than the other two methods. This indicates that when the experience pool is sequentially updated by two *TD-error* values of different orders of magnitude, the distribution characteristics between transitions will be destroyed, which easily leads agents to get stuck in local optima. As shown in Fig. 10(b), *#MADDPG-improved-PER* obtains the same high car-following reward as *#MADDPG* but with lower fuel consumption. This shows that using the average of two agents' *TD-errors* and halving the number of updates effectively alleviates the above problem and helps the *Agent EMS* learn better EMS. Table VI compares the three alternatives. Taking *#MADDPG* as the benchmark, the fuel cost per 100 km and overall driving cost of *#MADDPG-PER* are both more than 10% higher. And fuel consumption of *#MADDPG-improved-PER* is only 89.56% of *#MADDPG*, and the overall driving cost is 91.67%. It is noted that the overall driving cost includes fuel cost and battery degradation cost. The fuel price is 9.53 CNY/L, and the battery price is

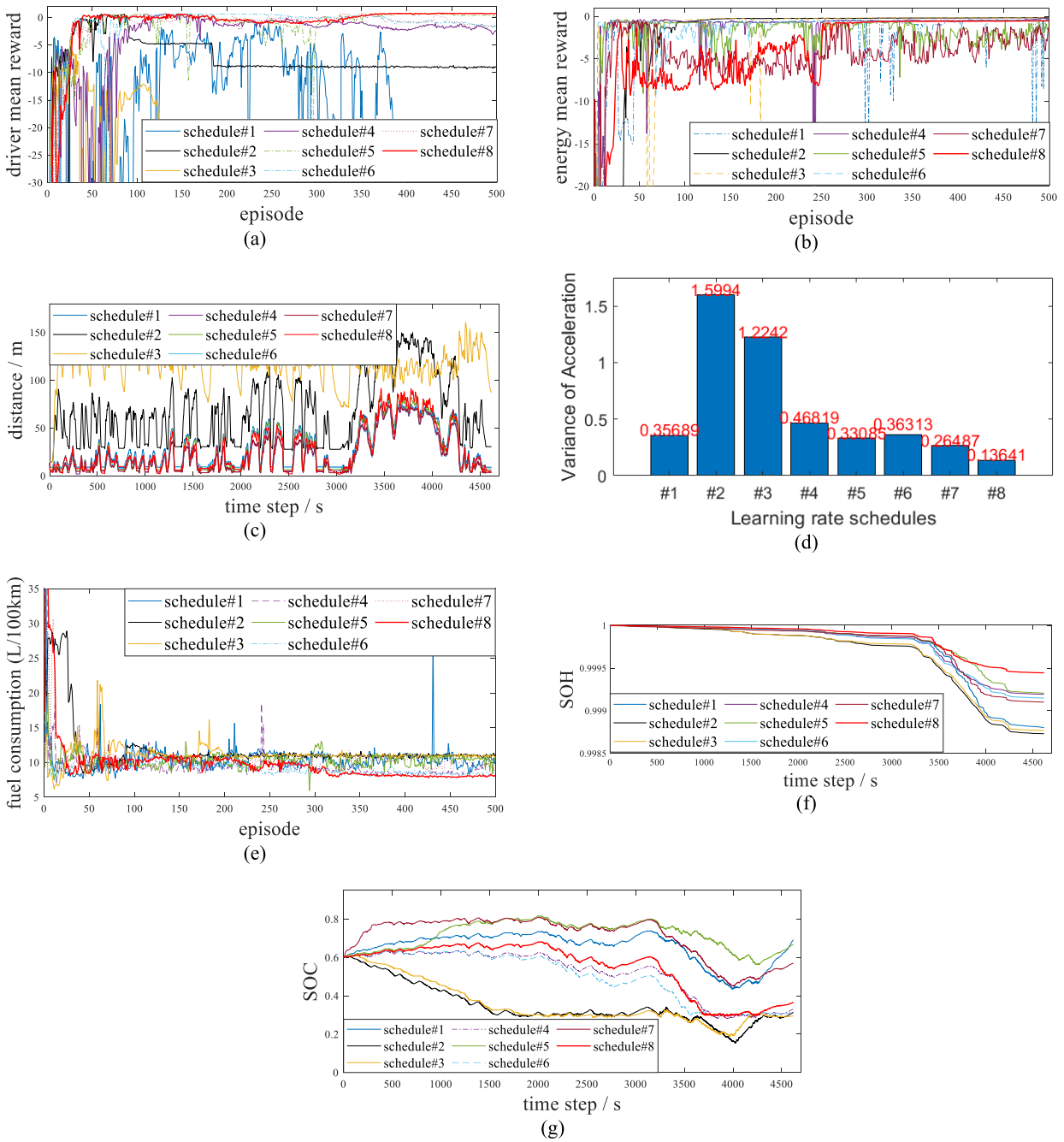

Fig. 9. Comparison of optimization results for different learning rate schedules. (a) Average reward of *Agent ACC* in each training episode. (b) Average reward of *Agent EMS* in each training episode. (c) Distance trajectory of the last training episode. (d) Variance of |acceleration| of the last training episode. (e) fuel consumption per 100 km in each training episode. (f) SOH trajectory of the last training episode. (g) SoC trajectory of the last training episode.

estimated to be 7420 CNY. The comparison shows that our improvement measure for PER has improved the optimization performance of the original MADDPG algorithm by more than 10%.

C. Analysis of Optimality

This section discusses the optimality of the proposed method from two aspects, car-following control and energy management. The single-agent DDPG algorithm is used in the collaborative optimization of ACC and EMS, the relevant discussions are published in our previous paper [35].

In terms of car-following performance, three car-following models, i.e., Krauss, Intelligent Driver Model (IDM), and ACC, are implemented in Simulation of Urban Mobility (SUMO). As shown in Fig. 11(a), the *#MADDPG-improved-PER* method learns a similar tracking trajectory to *#DDPG*, since they are both obtained through deep learning approaches. Compared with the models from SUMO, the proposed method has tighter car-following gaps, which means more stable and effective car-following performance, under the safety premise of noncollision. The variance of the absolute value of acceleration are listed in Table VII, whicht shows that the

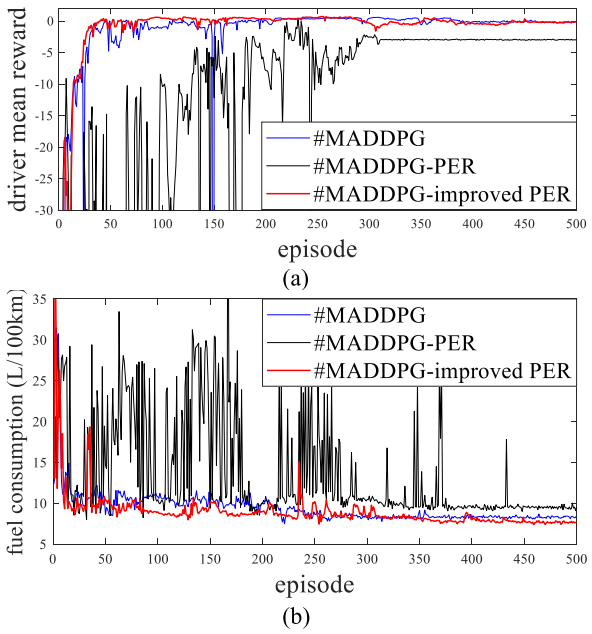

Fig. 10. Comparison of different experience replay techniques. (a) Average reward of Agent ACC in each training episode. (b) Fuel cost per 100 km in each training episode.

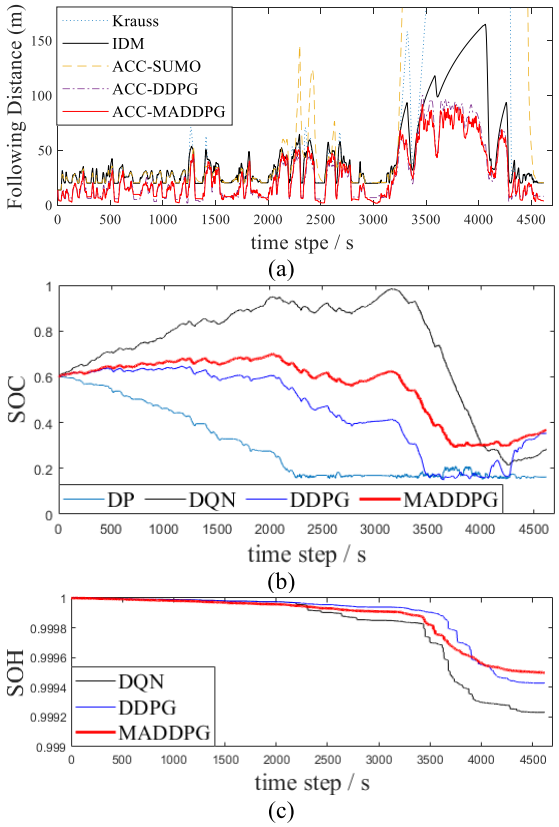

Fig. 11. Comparison of different methods and *#MADDPG-improved-PER*. (a) Distance trajectory of the last training episode. (b) SoC trajectory of the last training episode. (c) SoH trajectory of the last training episode.

#MADDPG-improved-PER method has the best comfortability performance among comparison experiments. In short, the proposed method outperforms car-following control.

In order to demonstrate superior performance in energy management optimization, besides DDPG, Deep Q-Network (DQN) and DP are implemented to train an EMS. And their driving cycles are all the same. The SoC trajectories in Fig. 11(b) illustrate a very effective energy management process. The *#DP* method tends to give priority to using a power battery to drive the vehicle, and makes SoC decrease evenly to finally keep it near preset value. Although the SoC trajectories based on DRL algorithm will converge to preset ranges, the change process is not as uniform as that of *#DP*. It is also in this process that different strategies are learned. The SoC trajectory of *#DQN* rises sharply in the early stage, and the change is very drastic, which also leads to poor health performance, as shown in Fig. 11(c). *#MADDPG* method has gentler SoC trajectory, and thus, better health performance. As the data listed in Table VIII, taking *#DP* as the global optimal benchmark, in terms of fuel consumption, *#DDPG* can reach 95.14% of *#DP*, while *#MADDPG-improved-PER* can reach 98.10% of *#DP*. Taking SoH into consideration, the overall cost of *#MADDPG-improved-PER*can reach 93.88% of *#DP*, which is also much better than *#DDPG* and *#DQN*. In summary, compared with other mainstream algorithms, the proposed method can achieve the best strategy in both car-following and energy management, proving that our method has superior eco-driving performance.

TABLE VII
COMPARISON OF COMFORTABILITY

Car-following method	Variance of absolute value of acceleration	Comparison
Krauss	0.1742	100%
IDM	0.1639	94.09%
ACC-SUMO	0.1474	84.62%
ACC-DDPG	0.1386	79.56%
ACC-MADDPG	0.1364	78.30%

TABLE VIII
COMPARISON OF FUEL COST AND OVERALL DRIVING COST

	#DP	*#DQN*	*#DDPG*	*#MADDPG-improved-PER*
Fuel cost (L/100km)	7.24	8.54	7.61	7.38
Comparison	100%	84.78%	95.14%	98.10%
Overall driving cost	CNY 72.99	CNY 82.66	CNY 79.90	CNY 77.75
Comparison	100%	88.30%	91.35%	93.88%

D. Verification of Adaptability

This section verifies the adaptability of the proposed method from two aspects. On the one hand, in order to prove that the learned strategy does not fall into local optimum due to fixed initial states (FIS), five groups of random initial states (RIS) are tested in the *Mix-train* cycle. $D_{h,l}$ and SoC are randomly sampled in [5], [20] m and [0.45, 0.7], respectively. As shown in Fig. 12(a), whether FIS or random initial state, the proposed strategy can realize car-following with safety and effectiveness and shows consistent tracking trajectories. The five groups of randomized tests showed the same level of comfortability

TABLE IX
COMPARISON OF COMFORTABILITY, FUEL CONSUMPTION AND OVERALL DRIVING COST

Groups	Variance of |acceleration|	Comparison	Fuel consumption (L/100km)	Comparison	Overall driving cost (¥)	Comparison
FIS	0.1364	100%	7.38	100%	77.75	100%
RIS #1	0.1269	93.04%	7.35	99.59%	74.58	95.92%
RIS #2	0.1525	111.80%	7.99	108.72%	81.28	104.54%
RIS #3	0.1375	100.81%	7.90	107.05%	79.23	101.90%
RIS #4	0.1438	105.43%	7.61	103.12%	75.90	97.62%
RIS #5	0.1511	110.78%	7.60	102.98%	77.19	99.28%

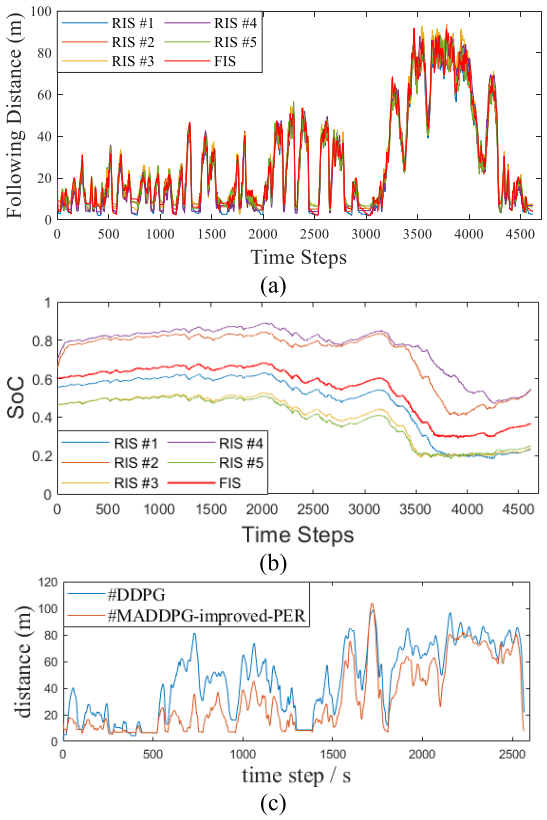

Fig. 12. Comparison of FIS and RIS. (a) Distance trajectory in *Mix-train* cycle. (b) SoC trajectory in *Mix-train* cycle. (c) Distance trajectory in *Mix-valid* cycle.

compared with the control group (FIS), the variance of the absolute value of acceleration does not exceed 11.8% upward and 6.9% downward, as the data listed in Table IX. The above results show that RIS do not worsen car-following performance. In terms of energy management performance, Fig. 12(b) shows similar SoC trajectories although the initial values are random. It can be seen that from Table IX, fuel consumption and overall driving cost of the five experimental groups (RIS) are not much different from those of the control group (FIS), and the differences between the two are controlled within 8.72% and 4.54%, respectively. The above results demonstrate that the proposed strategy will not fall into local optimum due to specific initial states; that is, the agent in different states can learn adaptive strategies that satisfy all the optimization objectives.

TABLE X
COMPARISON OF *#MADDPG-Improved-PER* IN *Mix-Train* AND *Mix-Valid* CYCLE

Driving cycle	Variance of |acceleration|	Fuel consumption (L/100km)	Overall driving cost (¥)	Comparison
Mix-train	0.1364	7.38	77.75	100%
Mix-valid	0.0883	7.67	74.48	95.79%

On the other hand, set the driving cycle of the leading car to *Mix-valid* of Fig. 7(b), and input it into the trained neural network model to verify the robustness of the proposed *#MADDPG-improved-PER* method. The distance curves shown in Fig. 12(c) illustrate that the proposed algorithm can also achieve good car-following performance on the verification cycle. And following distance of *#MADDPG-improved-PER* is less than *#DDPG*, which is noticeable in the mid-speed range. The comparison data in Table X shows that the ride comfort of *#MADDPG-improved-PER* on the validation cycle is much better than that in the training cycle; the variance of absolute value of acceleration is only 65% of the training cycle. More importantly, the overall driving cost of #MADDPG-improved-PER is also lower on the validation cycle, 95.79% of the training cycle. The above indicates that the proposed algorithm has good adaptability in urban and highway car-following conditions.

V. CONCLUSION

This article proposes an extensive eco-driving framework for HEVs, which can be decomposed into two subproblems, i.e., energy management and vehicle control in traffic scenarios. The MADDPG algorithm is employed to collaboratively optimize and synchronously control the eco-driving task, where two heterogeneous agents handle ACC and EMS, respectively. Thew main conclusions are as follows.

1) After explorations of learning rate schedules, the two heterogeneous agents can achieve Nash equilibrium during cooperation and competition, and the equilibrium points mean near-optimal performances. This will have reference significance for applications of the MADDPG algorithm in other fields.

2) Because of the improvement of PER technique, the proposed method has better fuel economy than the original MADDPG. The fuel consumption decreases by 10.44%, and the overall driving cost decreases by 8.42%.

3) The proposed method has superior ecological driving performance. It achieves more effective and comfortable tracking than the three car-following models in SUMO under the premise of safety. Compared to DP benchmark, its fuel consumption can reach 98.10% of that of DP, and overall driving cost can reach 93.88% of that of DP, which outperforms DQN and DDPG methods.

4) The proposed algorithm is adapted to different initial states, the differences in fuel consumption and overall driving cost are controlled within 8.72% and 4.54%, respectively, in check experiments.

5) The learned strategy is not sensitive to driving cycles. The overall driving cost of validation cycles is 95.79% of the training cycle. Consistent performances demonstrate that the proposed method has excellent adaptability in city, suburban, and highway driving conditions.

The proposed method may be challenging in learning rate adjustment because of games among multiple agents. In future work, we will deeply explore the eco-driving optimization problem in more real traffic scenarios so as to continue to improve the optimization performance through improving MADRL algorithms and reduce the difficulty of application.

REFERENCES

[1] F. U. Rui, Z. Ya-Li, and Y. Wei, "Progress and prospect in research on eco-driving," *China J. Highway Transp.*, vol. 32, no. 3, p. 1.

[2] O. D. Momoh and M. O. Omoigui, "An overview of hybrid electric vehicle technology," in *Proc. IEEE Vehicle Power Propuls. Conf.*, Sep. 2009, pp. 1286–1292.

[3] D. Shen, D. Karbowski, and A. Rousseau, "A minimum principle-based algorithm for energy-efficient eco-driving of electric vehicles in various traffic and road conditions," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 4, pp. 725–737, Dec. 2020.

[4] Z. Nie and H. Farzaneh, "Real-time dynamic predictive cruise control for enhancing eco-driving of electric vehicles, considering traffic constraints and signal phase and timing (SPaT) information, using artificial-neural-network-based energy consumption model," *Energy*, vol. 241, Feb. 2022, Art. no. 122888.

[5] J. Wei, G. Dong, and Z. Chen, "Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5634–5643, Jul. 2018.

[6] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng, "Eco-departure of connected vehicles with V2X communication at signalized intersections," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5439–5449, Dec. 2015.

[7] A. Panday and H. O. Bansal, "A review of optimal energy management strategies for hybrid electric vehicle," *Int. J. Veh. Technol.*, vol. 2014, pp. 1–19, Nov. 2014.

[8] J. Peng, H. He, and R. Xiong, "Rule based energy management strategy for a series–parallel plug-in hybrid electric bus optimized by dynamic programming," *Appl. Energy*, vol. 185, pp. 1633–1643, Jan. 2017.

[9] R. Lian, J. Peng, Y. Wu, H. Tan, and H. Zhang, "Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle," *Energy*, vol. 197, Apr. 2020, Art. no. 117297.

[10] S. G. Li, S. M. Sharkh, F. C. Walsh, and C. N. Zhang, "Energy and battery management of a plug-in series hybrid electric vehicle using fuzzy logic," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 3571–3585, Oct. 2011.

[11] R. M. Patil, J. C. Kelly, Z. Filipi, and H. K. Fathy, "A framework for the integrated optimization of charging and power management in plug-in hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 62, no. 6, pp. 2402–2412, Jul. 2013.

[12] S. Onori, L. Serrao, and G. Rizzoni, "Adaptive equivalent consumption minimization strategy for hybrid electric vehicles," in *Proc. ASME Dyn. Syst. Control Conf.*, Jan. 2010, pp. 499–505.

[13] L. Serrao, S. Onori, A. Sciarretta, Y. Guezennec, and G. Rizzoni, "Optimal energy management of hybrid electric vehicles including battery aging," in *Proc. Amer. Control Conf.*, Jul. 2011, pp. 2125–2130.

[14] S. Xie, J. Peng, and H. He, "Plug-in hybrid electric bus energy management based on stochastic model predictive control," *Energy Proc.*, vol. 105, pp. 2672–2677, May 2017.

[15] G. Jinquan, H. Hongwen, P. Jiankun, and Z. Nana, "A novel MPC-based adaptive energy management strategy in plug-in hybrid electric vehicles," *Energy*, vol. 175, pp. 378–392, May 2019.

[16] Q. Xue, X. Zhang, T. Teng, J. Zhang, Z. Feng, and Q. Lv, "A comprehensive review on classification, energy management strategy, and control algorithm for hybrid electric vehicles," *Energies*, vol. 13, no. 20, p. 5355, Oct. 2020.

[17] P. Zhang, F. Yan, and C. Du, "A comprehensive analysis of energy management strategies for hybrid electric vehicles based on bibliometrics," *Renew. Sustain. Energy Rev.*, vol. 48, pp. 88–104, Aug. 2015.

[18] S. Ebbesen, P. Elbert, and L. Guzzella, "Battery state-of-health perceptive energy management for hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 61, no. 7, pp. 2893–2900, Sep. 2012.

[19] L. Tang, G. Rizzoni, and S. Onori, "Energy management strategy for HEVs including battery life optimization," *IEEE Trans. Transport. Electrific.*, vol. 1, no. 3, pp. 211–222, Oct. 2015.

[20] Y. Li, H. He, J. Peng, and H. Zhang, "Power management for a plug-in hybrid electric vehicle based on reinforcement learning with continuous state and action spaces," *Energy Proc.*, vol. 142, pp. 2270–2275, Dec. 2017.

[21] Y. Wu, H. Tan, J. Peng, H. Zhang, and H. He, "Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus," *Appl. Energy*, vol. 247, pp. 454–466, Aug. 2019.

[22] X. Han, H. He, J. Wu, J. Peng, and Y. Li, "Energy management based on reinforcement learning with double deep Q-learning for a hybrid electric tracked vehicle," *Appl. Energy*, vol. 254, Nov. 2019, Art. no. 113708.

[23] J. Wu, H. He, J. Peng, Y. Li, and Z. Li, "Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus," *Appl. Energy*, vol. 222, pp. 799–811, Jul. 2018.

[24] Y. Zou, T. Liu, D. Liu, and F. Sun, "Reinforcement learning-based real-time energy management for a hybrid tracked vehicle," *Appl. Energy*, vol. 171, pp. 372–382, Jun. 2016.

[25] J. Wu, Z. Wei, W. Li, Y. Wang, Y. Li, and D. U. Sauer, "Battery thermal- and health-constrained energy management for hybrid electric bus based on soft actor-critic DRL algorithm," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 3751–3761, Jun. 2021.

[26] J. Wu, Z. Wei, K. Liu, Z. Quan, and Y. Li, "Battery-involved energy management for hybrid electric bus based on expert-assistance deep deterministic policy gradient algorithm," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12786–12796, Nov. 2020.

[27] Z. Gao and T. Laclair, "Electric and conventional vehicle performance over eco-driving cycles: Energy benefits and component loss," Oak Ridge Nat. Lab. (ORNL), Oak Ridge, TN, USA, Jul. 2019. [Online]. Available: https://www.osti.gov/biblio/1559604

[28] Z. Bai, P. Hao, W. ShangGuan, B. Cai, and M. J. Barth, "Hybrid reinforcement learning-based eco-driving strategy for connected and automated vehicles at signalized intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15850–15863, Sep. 2022.

[29] S. Li et al., "Overview of ecological driving technology and application for ground vehicles," *J. Automot. Saf. Energy*, vol. 5, no. 2, p. 121, 2014.

[30] S. Eben Li, H. Peng, K. Li, and J. Wang, "Minimum fuel control strategy in automated car-following scenarios," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 998–1007, Mar. 2012.

[31] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 348–368, Dec. 2018.

[32] G. Li and D. Görges, "Ecological adaptive cruise control and energy management strategy for hybrid electric vehicles based on heuristic dynamic programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3526–3535, Sep. 2019.

[33] S. K. Chada, J. M. Thomas, D. Görges, A. Ebert, and R. Teutsch, "Ecological adaptive cruise control for city buses based on hybrid model predictive control using PnG and traffic light information," in *Proc. IEEE Vehicle Power Propuls. Conf. (VPPC)*, Oct. 2021, pp. 1–7.

[34] M. Vajedi and N. L. Azad, "Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 113–122, Jan. 2016.

[35] J. Peng, Y. Fan, G. Yin, and R. Jiang, "Collaborative optimization of energy management strategy and adaptive cruise control based on deep reinforcement learning," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 1, pp. 34–44, Mar. 2023.

[36] S. Li, K. Li, R. Rajamani, and J. Wang, "Multi-objective coordinated control for advanced adaptive cruise control system," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly 28th Chin. Control Conf.*, Dec. 2009, pp. 3539–3544.

[37] Z. Zhu, N. Xie, K. Zong, and L. Chen, "Building a connected communication network for UAV clusters using DE-MADDPG," *Symmetry*, vol. 13, no. 8, p. 1537, Aug. 2021.

[38] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.

[39] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9895–9903, Oct. 2020.
[40] R. Z. Homod et al., "Dynamics analysis of a novel hybrid deep clustering for unsupervised learning by reinforcement of multi-agent to energy saving in intelligent buildings," *Appl. Energy*, vol. 313, May 2022, Art. no. 118863.
[41] S. M. Dawood, A. Hatami, and R. Z. Homod, "Trade-off decisions in a novel deep reinforcement learning for energy savings in HVAC systems," *J. Building Perform. Simul.*, vol. 15, no. 6, pp. 809–831, Nov. 2022.
[42] D. Yang, Y. Pu, F. Yang, and L. Zhu, "Car-following model based on optimal distance and its characteristics analysis," *J. Northwest Transp. Univ.*, vol. 47, no. 5, p. 888, 2012.
[43] Q. Luo, L. Xun, Z. Cao, and Y. Huang, "Simulation analysis and study on car-following safety distance model based on braking process of leading vehicle," in *Proc. 9th World Congr. Intell. Control Autom.*, Jun. 2011, pp. 740–743.
[44] Z. Liu, Q. Yuan, G. Nie, and Y. Tian, "A multi-objective model predictive control for vehicle adaptive cruise control system based on a new safe distance model," *Int. J. Automot. Technol.*, vol. 22, no. 2, pp. 475–487, Apr. 2021.
[45] Z. Wei, G. Dong, X. Zhang, J. Pou, Z. Quan, and H. He, "Noise-immune model identification and state-of-charge estimation for lithium-ion battery using bilinear parameterization," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 312–323, Jan. 2021.
[46] Z. Wei et al., "A noise-tolerant model parameterization method for lithium-ion battery management system," *Appl. Energy*, vol. 268, Jun. 2020, Art. no. 114932.
[47] X. Lin et al., "A lumped-parameter electro-thermal model for cylindrical batteries," *J. Power Sources*, vol. 257, pp. 1–11, Jul. 2014.
[48] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[49] R. Lowe et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.
[50] D. Silver, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 387–395.
[51] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.* Cham, Switzerland: Springer, 2017, pp. 66–83.
[52] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*.
[53] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472.
[54] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," 2020, *arXiv:2011.00583*.

Jiankun Peng received the Ph.D. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2016.

From June 2016 to November 2019, he served as a Postdoctoral Researcher with the National Engineering Laboratory of Electric Vehicles, Beijing Institute of Technology, Beijing. He is currently as an Associate Professor with the School of Transportation, Southeast University, Nanjing, China. He has more than ten years of research and working experience in modeling and control for new energy vehicles, where he has contributed more than 60 articles. His current research interests include energy management and optimization for electrified vehicles, connected and automated driving, and decision making for ecological driving.

Weiqi Chen was born in Hunan, China, in 1999. He received the B.S. degree in transportation engineering from Central South University, Changsha, China, in 2021. He is currently pursuing the M.S. degree with the School of Transportation, Southeast University, Nanjing, China.

His current research interests include multi-agent, deep reinforcement learning, eco-driving of hybrid electric vehicles, and automatic driving.

Yi Fan was born in Anhui, China, in 1998. He received the B.S. degree in transportation engineering from the Civil Aviation Flight University of China, Deyang, China, in 2020. He is currently pursuing the M.S. degree with the School of Transportation, Southeast University, Nanjing, China.

His current research interests include the optimal control and eco-driving of the hybrid electric vehicles.

Hongwen He (Senior Member, IEEE) received the M.S. degree from the Jilin University of Technology, Changchun, China, in 2000, and the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2003, both in vehicle engineering.

He is currently a Professor with the National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, Beijing Institute of Technology. His currently research interests include vehicle dynamics and control, power battery modeling, and simulation on electric vehicles, design and control theory of the hybrid power trains.

Zhongbao Wei (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees in instrumental science and technology from Beihang University, Beijing, China, in 2010 and 2013, respectively, and the Ph.D. degree in power engineering from Nanyang Technological University, Singapore, in 2017.

He was a Research Fellow with Energy Research Institute, Nanyang Technological University, from 2016 to 2018. He is currently a Professor in vehicle engineering with the National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, Beijing Institute of Technology, Beijing. He has authored or coauthored more than 40 peer-reviewed articles. His research interests include modeling, identification, state estimation, diagnostic for battery, and energy management for hybrid energy systems.

Chunye Ma received the M.Sc. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1999 and 2004, respectively, both in mechanical engineering.

He is currently a National Distinguished Expert, working as a Chief Professor with the School of Transportation, Southeast University, Nanjing, China. He has more than 30 years of working experience in international automotive companies, including engineering and management experience at Ford Motor, Dearborn, MI, USA and Groupe PSA, Rueil-Malmaison, Paris, France. He was involved in strategic product planning, vehicle project and supplier management, manufacturing, and quality control. His current research interests include intelligent transportation, smart city, vehicle-city integration, vehicle-road collaboration, and intelligent Internet-connected vehicle.