ENERGY & SYSTEM
Research Article **|** ***Volume 5*** **|** ***Issue 5*** **|** ***2025***
Received: 6 Jan 2025 | *Revised: 19 Jan 2025*
Accepted: 23 Jan 2025 | *Published Online: 25 Jan 2025*

Predictive Energy Management Strategy for Hybrid Electric Vehicles Based on Soft Actor-Critic

Weidong Huang[1], **Jiahuai Ma** [2,*]

1 Upower Energy Technology (Guang Zhou) Co..Ltd., Guangzhou, 510000, CHINA

2 Corresponding, University of Florida, Herbert Wertheim College, FL, 32608, USA,
Email: maj2@ufl.edu

Abstract: This paper presents a predictive Energy Management Strategy (EMS) for series hybrid electric vehicles based on an improved Soft Actor-Critic (SAC) algorithm. First, the Informer model is used to predict the vehicle's short-term speed trajectory, providing foresight to guide the optimization of the energy management strategy. Second, by incorporating a prioritized experience replay strategy, the convergence of the SAC algorithm is accelerated, and its performance is enhanced. Finally, a simulation environment based on real driving cycles was constructed, and the simulation results demonstrate that our method effectively reduces fuel consumption, achieving approximately a 6.1% performance improvement over the original SAC algorithm. This not only validates the superiority of our approach over traditional methods in terms of fuel efficiency but also provides new insights into energy management for hybrid electric vehicles.

Keywords: *Hybrid electric vehicles, Energy management, Vehicle speed prediction, Reinforcement learning*

1. Introduction

With the rapid development of the global automotive industry, energy shortages and environmental pollution have become two major bottlenecks restricting sustainable development. The traditional dependency of fuel-powered vehicles on oil has not only accelerated the depletion of limited energy resources but has also triggered a series of severe environmental challenges, such as global warming caused by greenhouse gas emissions and deteriorating air quality from harmful exhaust pollutants [1][2]. Against this backdrop, Hybrid Electric Vehicles (HEV), which integrate both fuel and electric propulsion, have emerged as a new transportation solution and are widely regarded as a critical technological transition from conventional gasoline-powered vehicles to fully electric vehicles [3][4][5].

HEV typically consist of two or more power sources, making the energy management system an indispensable component. With appropriate control strategies, HEV can efficiently operate by coordinating multiple power sources, thereby reducing fuel consumption and greenhouse gas emissions. Generally, energy management strategies for HEV can be broadly classified into three categories: rule-based approaches, optimization-based approaches, and learning-based approaches.

Rule-based methods include deterministic and fuzzy rule-based strategies. These methods are widely applied in HEVs due to their simplicity and real-time performance advantages. Li et al. [6] proposed a novel Q-learning strategy based on deterministic rules for real-time energy management in HEV, which meets the driver's traction demands while reducing fuel consumption and load fluctuations and enhancing adaptability to different driving cycles. Phillips [7] designed a supervisory controller for energy management. By analyzing various operational modes of the vehicle and dynamic control strategies, a logical structure was constructed to guide the smooth transition between different modes. Lv et al. [8] introduced a fuzzy control-based energy management strategy for plug-in parallel HEV. This strategy, building upon rule-based algorithms, employs fuzzy control for smoother control effects, effectively mitigating the rate of decline in the battery state of charge (SOC) and significantly reducing fuel consumption. Gao et al. [9] proposed a power management strategy based on a fuzzy logic controller, optimizing the hybrid degree and membership functions using the golden ratio cutoff rule to achieve the optimal power distribution between batteries and supercapacitors. However, further applications are hindered by limited optimality and dependence on expert knowledge, while preset rules constrain flexibility under varying driving conditions.

Optimization-based energy management strategies can be categorized into global optimization and real-time optimization. Global optimization includes Dynamic Programming (DP), Genetic Algorithms (GA), and convex optimization; real-time optimization encompasses Model Predictive Control (MPC), Pontryagin’s Minimum Principle (PMP), and Equivalent Consumption Minimization Strategies (ECMS). Tang et al. [10] proposed an improved dynamic programming algorithm capable of accurately identifying regions containing multiple optimal state-of-charge trajectories, thereby reducing computational complexity while ensuring fuel economy. Shi et al. [11] drew on dynamic programming strategies to design a reference SOC curve and adaptive adjustment mechanism, allowing the SOC to linearly decrease with driving distance and reach a minimum value at the end of the trip. Farajpour et al. [12] used experimental methods to ascertain the characteristics of built-in Permanent Magnet Synchronous Motors and simulated the motor inverter system using artificial neural networks, calculating the kinetic energy required to drive the vehicle based on a longitudinal vehicle model. Genetic algorithms were utilized to determine optimal operational criteria for minimizing force required at the axle and power losses in electronic devices. Li et al. [13] emphasized the crucial role of convex optimization in electric vehicle design and control, summarizing various convex optimization methods used for component sizing and energy management, and discussing their application prospects in enhancing electric vehicle efficiency and reducing costs. Zhang et al. [14] proposed an improved adaptive equivalent minimization strategy that adjusts the equivalent factor by predicting future driving conditions to enhance the fuel economy of plug-in hybrid vehicles. Although these methods provide avenues for optimizing vehicle performance under specific conditions, the high computational costs of these EMSs, as well as their limited adaptability to complex driving cycles, hinder the attainment of optimal solutions. In recent years, learning-based approaches have offered more optimal solutions for EMS, primarily through Reinforcement Learning (RL) algorithms that present an alternative solution to challenging control problems in both virtual and real-world environments. Research in the area of energy management for HEV has also shown that reinforcement learning exhibits strong learning capabilities and adaptability under complex driving cycles while consuming fewer computational resources [15].

Zhou et al. [16] proposed a new model-free multi-step reinforcement learning EMS featuring three multi-step learning strategies: sum to terminal, average to neighbors, and cycle to terminal. Compared to well-designed model-based predictive energy management control strategies, the proposed energy management method could increase predictive length by 71% in hardware-in-the-loop experiments and save at least 7.8% of energy under identical driving conditions. Additionally, Liu et al. [17] designed a Q-learning-based adaptive energy management method for hybrid electric tracked vehicles. Results demonstrated that this approach exhibited stronger adaptability, optimality, and learning capability than stochastic dynamic programming, effectively reducing computation time. Xiong et al. [18] utilized the same algorithm to achieve optimal power distribution between the battery and supercapacitor in plug-in HEV, significantly reducing energy loss by 16.8%. Wu et al. [19] employed a Deep Q-Learning (DQL) algorithm for power distribution, which not only addressed the curse of dimensionality encountered in Q-learning but achieved better fuel economy. Wang et al. [20] proposed an energy management strategy based on a mutation-protected deep Q-network (MPD) designed to enhance hydrogen economy in fuel cell vehicles and reduce fuel cell degradation. By quantifying the mutations in driving cycles and combining them with driving conditions, the MPD-EMS achieved approximately 11% and 6% reductions in hydrogen consumption compared to other learning-based EMSs, as well as about 21% and 13% reductions in fuel cell degradation. Compared to DQL, further exploration by Wu and Tan et al. [21][22] revealed that employing the Deep Deterministic Policy Gradient (DDPG) algorithm with continuous state and action representations, along with prioritized experience replay algorithms, improved EMS learning efficiency, yielding results nearly equivalent to DP performance.

Moreover, building on the aforementioned studies, the development of vehicle intelligence and connectivity technologies has made incorporating vehicle speed prediction into energy management strategies a focal point of research. Vehicle speed prediction can provide anticipatory information for energy management strategies by estimating future driving conditions, thereby optimizing power source allocation and enhancing fuel economy and emissions performance. Sun et al.[23] proposed a Markov chain-based speed prediction method, which created a speed state transition matrix to predict future speeds over a short period, subsequently applied to energy management strategies for dynamic adaptation to changes in driving conditions. Zhang et al. [24] employed an autoregressive integrated moving average (ARIMA) model to forecast future vehicle speeds, integrating this data with dynamic programming to optimize HEV energy management strategies and improve fuel economy. Liu et al. [25] constructed a vehicle speed prediction model using Support Vector Machines (SVM) while optimizing energy distribution with Model Predictive Control (MPC), significantly enhancing vehicle operational efficiency. Chen et al. [26] utilized Long Short-Term Memory (LSTM) networks to process time-series data for vehicle speeds, establishing a real-time prediction model that provided accurate future speed sequences, thereby optimizing HEV energy management performance. Wang et al. [27] introduced a traffic-cooperative speed prediction method that utilized traffic signal states and preceding vehicle information to anticipate future speeds, successfully reducing fuel consumption in conjunction with a rule-optimized energy management strategy. Ding et al. [28] developed a future speed prediction model based on random forest algorithms by integrating vehicular network information, achieving a more efficient energy allocation strategy through predictive insights. Wang et al. [29] integrated an LSTM speed prediction model with Deep Reinforcement Learning (DRL) algorithms to design an end-to-end energy management framework, wherein future speed information provided by the prediction module significantly improved the strategy's anticipatory and robust performance. Zou et al. [30] designed a fuzzy logic-based predictive controller incorporating future speed as an input variable, effectively enhancing HEV energy management adaptability under complex driving conditions.

However, existing research presents room for improvement in the integration of speed prediction information with reinforcement learning algorithms, particularly regarding the construction of reinforcement learning state spaces and enhancing algorithm learning efficiency. The main contributions of this paper are as follows:

- We design a vehicle speed prediction module that integrates predicted speed sequences into the reinforcement learning state space, providing anticipatory traffic information to facilitate proactive energy allocation strategy planning, achieving predictive performance in energy management.
- We propose a hybrid vehicle energy management strategy based on the SAC algorithm and develop a prioritized experience replay mechanism to enhance the learning efficiency and performance of the reinforcement learning algorithm.
- We conduct simulation experiments under real driving cycles, with experimental results demonstrating that the proposed method surpasses classical baseline models in controlled experiments.

2. Modeling of Series Hybrid Electric Vehicle

This study focuses on the study of Series Hybrid Electric Vehicles (SHEV), with the power system structure illustrated in Figure 1. The system primarily consists of an internal combustion engine, generator, battery pack, electric motor, and main reduction gear. The internal combustion engine drives the generator to produce electrical energy, which can be utilized directly to power the electric motor or stored in the battery. The electric motor is ultimately responsible for delivering energy to the wheels to propel the vehicle. The battery pack serves not only to store regenerative braking energy and excess energy but also to allow charging from an external power source. This structure effectively decouples the internal combustion engine from the drive wheels, simplifying the control strategy of the powertrain.

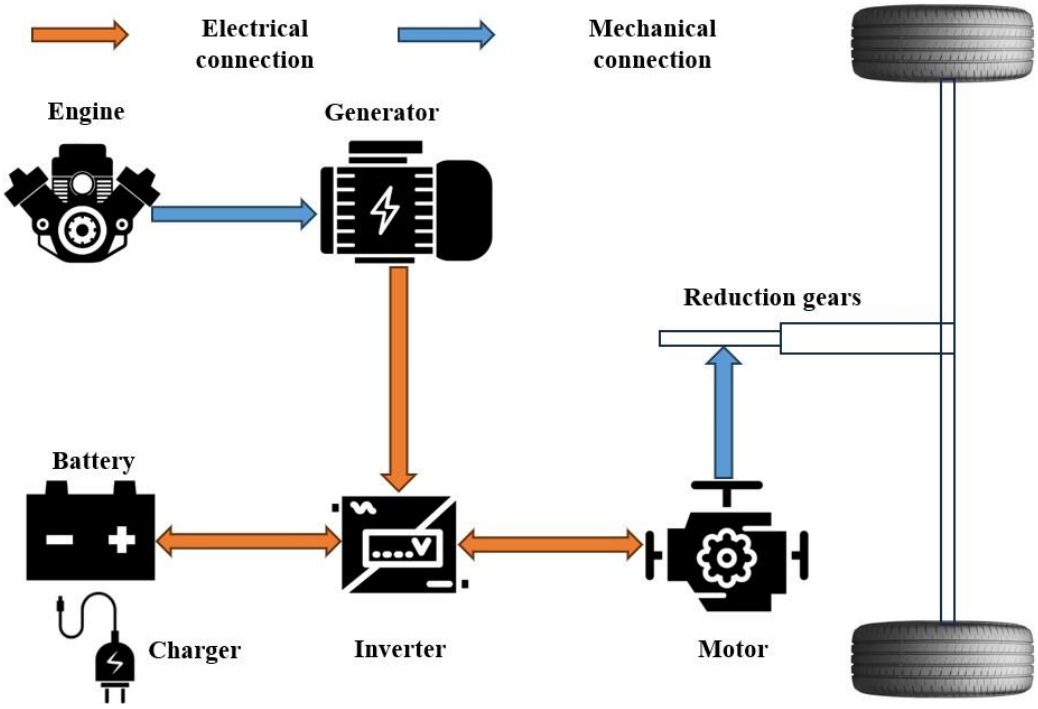

Figure 1 Power System Structure of SHEV

A. Powertrain Model of Series Hybrid Electric Vehicle

This study analyzes a mid-size passenger vehicle, with its key parameters listed in Table 1. As illustrated in Figure 1, the energy supply for the electric traction motor is derived from two sources: the engine-generator set (EGS) and the lithium-ion battery (LIB) pack. Therefore, the powertrain of the SHEV can be divided into two sub-models: the engine-generator set model and the lithium-ion battery model. The following sections will elaborate on these two sub-models in detail.

Table 1: Key Parameters of the Series Hybrid Electric Vehicle (SHEV) Specifications

Symbol	Parameter	Value
P_{eng}^{max}	Peak power of the gasoline engine	56 kW
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	2100 kg
$A_{)}$	Windward area	2.42 m^2
$R_{)h}$	Wheel radius	0.287 m
i_0	Main reducer ratio	3.64
C_n	Nominal capacity of battery pack	7.42 kWh

B. Engine-Generator Set (EGS) Model

This study focuses on analyzing the role of the engine-generator set (EGS) and the lithium-ion battery (LIB) in the energy distribution process, assuming that the electric motor's driving force is uniformly distributed between the front and rear axles. The energy demand of the motor depends on the vehicle's mass, speed, and acceleration, which are critical to the energy management system's energy allocation function. Under given vehicle speed v and acceleration a, the total power demand P_{req} can be expressed as:

$$P_{req} = v \cdot F_{total} \tag{1}$$

where F_{total} represents the total resistance experienced by the vehicle during motion, including inertial resistance, rolling resistance, grade resistance, and aerodynamic drag, computed as follows:

$$\begin{gathered} F_{total} = F_a + F_r + F_i + F_{)} \\ F_a = m \cdot a \\ F_r = \mu mg\cos\,\theta \\ F_i = mg\sin\,\theta \\ \boldsymbol{F}_{)} = \frac{A_{)}C_d v^2}{21.15} \end{gathered} \tag{2}$$

here, F_a is the inertial resistance determined by mass m and acceleration a; F_r is rolling resistance, with μ assumed to be 0.01; F_i is the grade resistance, with θ being the road slope angle (assumed to be 0 in this study); $F_{)}$ represents aerodynamic drag, where $A_{)}$ is the windward area and C_d is the drag coefficient (set to $C_d = 0.65$ for this study); gravitational acceleration g is taken as 9.8 m/s^2.

Assuming that the engine-generator set (EGS) can respond rapidly after receiving a control signal, the energy conversion between the engine and the generator can be described through a quasi-static fuel consumption model and power transmission model, with efficiency obtained from efficiency maps. The relationship between torque and speed between the engine and the generator is represented as follows:

$$T_{eng} = T_{gen}, W_{eng} = W_{gen} \tag{3}$$

Based on the current torque and speed, the output power of the engine and generator can be computed using the following equations:

$$\begin{gathered} P_{eng} = T_{eng} \cdot W_{eng} \\ P_{gen} = T_{gen} \cdot W_{gen} \cdot \eta_{gen} \end{gathered} \tag{4}$$

where η_{gen} denotes the efficiency of the generator.

The fuel consumption rate can be calculated using the lower heating value G and the engine's efficiency η_{eng} as follows:

$$\dot{m}_f = \frac{P_{eng}}{G \cdot \eta_{eng}} \tag{5}$$

where $G = 4.25 \times 10^{3}\mathrm{J/kg}$.

The torque and speed of the engine and generator must satisfy the following boundary conditions:

$$\begin{gathered} T_{eng}^{min} \leq T_{eng} \leq T_{eng}^{max}, T_{gen}^{min} \leq T_{gen} \leq T_{gen}^{max}, \\ W_{eng}^{min} \leq W_{eng} \leq W_{eng}^{max}, W_{gen}^{min} \leq W_{gen} \leq W_{gen}^{max}. \end{gathered} \tag{6}$$

The electric traction power is jointly supplied by the generator and the lithium-ion battery, considering the inverter efficiency η_{inv}. The total power demand can thus be expressed as:

$$P_{req} = (P_{bat} + P_{gen}) \cdot \eta_{inv}. \tag{7}$$

where P_{bat} is the power provided by the lithium-ion battery, P_{gen} is the power supplied by the generator, and η_{inv} denotes the inverter's efficiency, assuming a fully regenerative braking strategy.

The aforementioned model enables the energy distribution between the engine-generator set and the battery, providing theoretical support for the energy management strategies of hybrid systems.

C. Lithium-Ion Battery (LIB) Model

The lithium-ion battery (LIB) employs a coupled electro-thermal-aging model for simulation, comprising three sub-models: a second-order RC circuit model, a dual-state thermal model, and an energy flux aging model. The circuit model is coupled with the thermal model to accurately describe the electrothermal dynamic characteristics of the LIB. Within the circuit model, the voltage source represents the open-circuit voltage (OCV), which is related to the battery's SOC, while the total ohmic resistance R_s represents the battery's equivalent internal resistance. Furthermore, LIB operation is influenced by polarization effects, including charge transfer, diffusion phenomena, and the effects of the passivation layer on the electrodes. To simulate these phenomena, two RC branches are used in the modeling. The governing equations for the circuit sub-model are as follows:

$$\frac{dSoC(t)}{dt} = \frac{I(t)}{3600C_n} \tag{8}$$

$$\frac{dV_{p1}(t)}{dt} = -\frac{V_{p1}(t)}{R_{p1}(t)C_{p1}(t)} + \frac{I(t)}{C_{p1}(t)} \tag{9}$$

$$\frac{dV_{p2}(t)}{dt} = -\frac{V_{p2}(t)}{R_{p2}(t)C_{p2}(t)} + \frac{I(t)}{C_{p2}(t)} \tag{10}$$

$$V_t(t) = V_{oc}(SoC) + V_{p1}(t) + V_{p2}(t) + R_sI(t) \tag{11}$$

where $I(t)$ and $V_t(t)$ represent the current and terminal voltage of the battery at time t respectively; V_{p1} and V_{p2} are the polarization voltages of the two RC branches; C_{p1} and C_{p2} are the capacitances of the polarization branches, while R_{p1} and R_{p2} are the resistances of the branches; V_{oc} is the open-circuit voltage, dependent on SOC; and R_s is the total equivalent internal resistance.

According to the principle of thermal energy conservation, the temperature dynamics of the LIB can be described by the following thermal balance equations:

$$C_c\frac{dT_c(t)}{dt} = \frac{T_s(t) - T_c(t)}{R_c} + H(t) \tag{12}$$

$$C_s\frac{dT_s(t)}{dt} = \frac{T_c(t) - T_s(t)}{R_c} + \frac{T_f(t) - T_s(t)}{R_u} \tag{13}$$

$$T_a(t) = \frac{T_c(t) + T_s(t)}{2} \tag{14}$$

here, T_s, T_c, T_a, and $T_f(t)$ represent the surface, core, average, and ambient temperatures of the battery (in °C), respectively; R_c and R_u denote the internal thermal resistance of the battery and the thermal resistance between the battery surface and the environment, respectively; C_c and C_s refer to the equivalent thermal capacities of the battery core and surface; and $H(t)$ denotes the rate of heat generation within the LIB, composed of ohmic heat, polarization heat, and irreversible entropy heat. The heat generation rate can be calculated as follows:

$$H(t) = I(t)\mathrm{V}V_{p1}(t) + V_{p2}(t) + R_sI(t)\mathrm{W} + I(t)[T_a(t) + 273]E_n(\mathrm{SOC}, t) \tag{15}$$

where E_n represents the heat generated from the entropy change during the electrochemical reaction.

Additionally, referencing the battery degradation assessment model based on energy flux proposed by Ebbesen et al.[31], the degradation characteristics of lithium-ion batteries under long-term charge-discharge cycles are analyzed. It is assumed that the battery can withstand a certain cumulative charge flow before reaching the end of its life, leading to the introduction of a dynamic evolution equation for the battery's state of health (SOH), expressed as:

$$\frac{dSOH(t)}{dt} = -\frac{\int_0^t |I(\tau)|d\tau}{2N(c, T_a)C_n}, \tag{16}$$

discretizing the above expression yields:

$$\Delta SOH_t = -\frac{|I(t)|\Delta t}{2N(c, T_a)C_n} \tag{17}$$

where $N(c,\ T_a)$ is the equivalent charge-discharge cycles until the end of the battery's life, C_n is the rated capacity of the battery, and Δt is the duration of charge-discharge.

At the same time, the capacity degradation of the battery is influenced by the rate c and the operating temperature T_a. Based on the Arrhenius equation, the capacity loss of the battery can be calculated as:

$$\Delta C_n = B(c) \cdot \exp \text{ k}-\frac{E_a(c)}{RT_a}\text{l} \cdot Ah^z \tag{18}$$

where ΔC_n denotes the percentage of capacity loss; $B(c)$ is the pre-exponential factor related to the rate, indicating the factors leading to Table 2; $E_a(c)$ is the activation energy; R is the ideal gas constant （8.314 J/mol·K); z is the exponent factor; and Ah is the cumulative charge flow of the battery.

Table 2: Dependence of Pre-exponential Factor on C-rate

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

The activation energy E_a can be expressed as:

$$E_a(c) = 31700 - 370.3 \cdot c. \tag{19}$$

When the battery's capacity C_n falls to 80% of its initial capacity, based on this definition and the definition of ΔSoH_t, the values of Ah and N can be derived as follows:

$$Ah(c,T_a) = \text{o}\frac{20}{B(c)} \cdot \exp \text{p}-\frac{E_a(c)}{RT_a}\text{qr}^{\frac{1}{z}}, \tag{20}$$

at this point, the battery reaches its end of life (EOL), and the corresponding number of charge-discharge cycles is given by:

$$N(c,T_a) = 3600 \cdot Ah(c,T_a)/C_n \tag{21}$$

Finally, the change in SOH under specific current, temperature, and dynamic operating conditions can be calculated using Equation (17) to assess the aging status of the battery pack.

Through the coupled simulation of the circuit, thermal, and aging models, the electrothermal characteristics and aging mechanisms of lithium-ion batteries can be accurately described, providing theoretical support for monitoring and predicting battery longevity.

3. Methods

The predictive EMS framework proposed in this paper is illustrated in Figure 2. It includes components for vehicle speed prediction using the Informer model, the application of the SAC algorithm, integration of a prioritized experience replay strategy, and the design of the SAC training strategy.

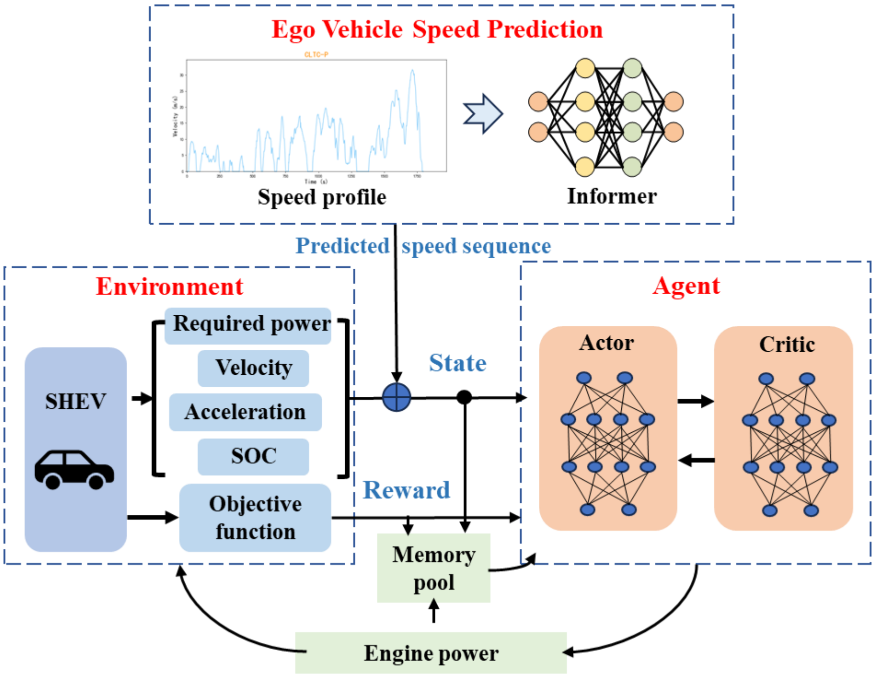

Figure 2 Overall Architecture of the EMS for SHEV

3.1 Vehicle Speed Prediction Based on Informer

Vehicle speed prediction is a significant research issue in the field of intelligent driving. The objective is to accurately predict future speeds based on the vehicle's historical speed data. This problem holds considerable importance for autonomous driving systems, intelligent traffic management, and eco-driving. Accurate speed prediction can provide better driving decision support, enhancing safety and economic efficiency.

In this work, we employ the Informer model for vehicle speed prediction. The Informer is an efficient time series prediction model based on an attention mechanism, which significantly improves the efficiency and accuracy of long-term predictions through the design of ProbSparse Self-Attention and a generative decoder. Compared to traditional time series prediction models such as LSTM and GRU, the Informer can handle long-term dependency issues and achieves significant advantages in computational complexity.

3.1.1 Problem Description

At the current time step t, the historical speed data set for the vehicle can be represented as:

$$X = \{v_t, v_{t-1}, \dots, v_{t-L+1}\} \in \mathbb{R}^L \tag{22}$$

where v_t denotes the speed of the vehicle at time t, and L represents the length of the historical observation window.

The goal is to predict the speed data set for the next T time steps:

$$Y = \{v_{t+1}, v_{t+2}, \dots, v_{t+T}\} \in \mathbb{R}^T \tag{23}$$

By training the Informer model, we aim to learn the mapping relationship between the input historical speed data X and the output future speed data Y:

$$Y = f_{\text{Informer}}(X; \theta) \tag{24}$$

where θ represents the model parameters.

In this way, the model can learn the spatial and temporal characteristics of speed changes and capture the patterns of speed evolution across different time scales.

3.1.2 Model Structure

The Informer model mainly consists of two components: ProbSparse Self-Attention and a generative decoder. ProbSparse Self-Attention optimizes the attention computation mechanism, while the generative decoder enhances decoding efficiency through parallel computation. Specifically, in ProbSparse Self-Attention, the traditional self-attention mechanism of the Transformer model exhibits high computational complexity when processing long sequential data. The Informer reduces computational complexity by sparsifying the self-attention mechanism, prioritizing significant attention distributions.

In the traditional self-attention mechanism, the attention scores are calculated as follows:

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{25}$$

where $Q \in \mathbb{R}^{(L_g \times d_k)}$ is the query matrix, L_q is the query sequence length, d_k is the dimensionality of the keys; $K \in \mathbb{R}^{(L_k \times d_k)}$ is the key matrix, L_k is the key sequence length; and $V \in \mathbb{R}^{(L_k \times d_v)}$ is the value matrix, with d_v being the value dimensionality.

This formula has a computational complexity of $O(L_q \cdot L_k \cdot d_k)$. When both L_q and L_k are large (e.g., in long sequence data), the computational expense becomes very high. To reduce this computational complexity, the Informer introduces ProbSparse Self-Attention, which retains only the few attention scores that have the greatest influence on the final output. The underlying principle is based on the observation that the attention distribution is often sparse, meaning that most query points focus on only a few key points, with the remaining attention scores contributing little to the output. Specifically, ProbSparse Self-Attention utilizes importance score filtering and sparse attention computation for optimization.

To identify the key points that have the greatest influence on the query points, the Informer defines the importance score U_i for each query point q_i:

$$U_i = \| q_i \|_2 \tag{26}$$

where $\|q_i\|_2$ denotes the L2 norm of the query point. By sorting U_i, only the maximum subset corresponding to the key points k_j is selected to construct the sparse attention matrix.

Next, the attention scores are calculated based on the selected important key points:

$$\mathrm{SparseAttention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK_{\mathrm{sparse}}^T}{\sqrt{d_k}}\right)V_{\mathrm{sparse}} \tag{27}$$

where K_{sparse} and V_{sparse} are the filtered key and value matrices.

Through the probabilistic sparse mechanism, the complexity of the attention computation decreases from the traditional $O(L_q \cdot L_k \cdot d_k)$ to $O(L_q \cdot \log L_k \cdot d_k)$. This optimization primarily benefits from two factors: the filtering step using U_i that computes only significant attention scores, and the sparsification of attention distributions, which significantly reduces unnecessary computations.

In the generative decoder component, traditional decoders typically employ a step-by-step decoding strategy. This means the model predicts one future value at each time step, using the predicted value as input for the subsequent time step. This approach is less efficient and prone to cumulative error. To address this, the Informer introduces a generative decoder that produces multiple predictions for future time steps in a single pass, significantly enhancing decoding efficiency.

In the stepwise prediction of traditional decoders, if we want to predict data for the next T time steps $Y\hat{} = \{v\hat{}_{t+1}, v\hat{}_{t+2}, \ldots, v\hat{}_{t+T}\}$, the prediction process of the traditional decoder is carried out incrementally:

$$v\text{å}_{t+i} = f_{\text{Decoder}}(v\text{å}_{t+1}, \text{ẫ}_{t+2}, \ldots, \text{ẫ}_{t+i-1}, X; \theta) \quad (28)$$

where $f_{Decoder}$ is the decoder model; X is the historical input data; θ are the model parameters; and each predicted value $v\hat{}_{t+i}$ serves as input for the next step until all T predictions are completed.

This stepwise prediction method presents two key issues: first, it is inefficient, as each step's prediction depends on the previous one, leading to T cycles and a high time complexity; secondly, it is susceptible to cumulative errors, where the prediction error of each step affects the input for the next, causing the error to amplify progressively.

By improving the generative decoder to generate predictions for all T future time steps at once, we can avoid redundant calculations and cumulative error issues prevalent in stepwise predictions. The improved generative decoder's time complexity is $O(T \cdot d)$, where T is the number of future time steps and d is the feature dimension. In contrast, the traditional decoder's time complexity is $O(T^2 \cdot d)$ due to the repeated calculations in each time step, thus providing a notable speed advantage for the generative decoder when handling long time sequences, while also eliminating cumulative error problems stemming from earlier-step inaccuracies, allowing for better global pattern focus and enhancing prediction accuracy and stability. The core formula for the generative decoder is:

$$\text{Ɏ} = f_{\text{Decoder}}(X; \theta) \quad (29)$$

where $Y\hat{} = \{v\hat{}_{t+1}, v\hat{}_{t+2}, \ldots, v\hat{}_{t+T}\}$ denotes the model's predictions for the next T time steps, generated all at once; X represents the historical input data, indicating the vehicle's historical speed sequence $\{v_t, v_{t-1}, \ldots, v_{t-L+1}\}$; and θ refers to the model parameters.

The generative decoder directly inputs historical data X into the decoder to generate the complete forecast sequence in one go, thus avoiding the repeated computations associated with stepwise decoding. The generative decoder is designed with a global attention mechanism to capture global temporal patterns within the historical input data X and apply these patterns directly to future time step predictions without relying on sequential inputs. Specifically, the decoder's output can be expressed as follows:

$$H = \text{Attention}(Q, K, V) \quad (30)$$

where $Q = X_{future}$ is the query vector for future time steps; K and V are the key and value matrices of the historical input data; and H is the hidden feature representation generated for the future time steps.

Furthermore, a parallel prediction mechanism is employed to directly generate the feature matrix H for all future time steps via parallel computation, followed by a mapping function to produce the final predicted values:

$$\text{Ɏ} = WH + b \quad (31)$$

where W and b are the linear mapping weights of the decoder; $Y\hat{}$ is the prediction result generated in one pass by the decoder.

3.2 Soft Actor-Critic Algorithm

Reinforcement learning (RL) is a method for learning optimal decision-making policies through the interaction between an agent and its environment. Numerous studies have demonstrated the advantages of reinforcement learning in energy management for Hybrid Electric Vehicles (HEV), showcasing its significant application potential. The SAC algorithm emphasizes maximizing the entropy of the policy (i.e., the randomness of actions) while achieving the maximum reward based on the maximum entropy principle. This addresses the overestimation problem faced by traditional reinforcement learning algorithms like DDPG, thereby enhancing the model's exploration ability and decision robustness. SAC is particularly suited for continuous control tasks.
The objective function for SAC is defined as:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot\,|s_t))], \tag{32}$$

where $r(s_t, a_t)$ represents the reward function; $\mathcal{H}(\pi(\cdot\,|s_t))$ denotes the entropy of the policy; and α is the entropy regularization coefficient that balances the relationship between reward and entropy.

The SAC algorithm primarily comprises several core modules: the policy network, value network, and automatic temperature adjustment.

(1) **Policy Network (Actor Network)**.
The policy network outputs the parameters of the conditional distribution and generates continuous actions through Gaussian distributions. The objective of the policy is to maximize the cumulative reward that is regularized by entropy. The policy network functions to generate the probability distribution of actions taken by the agent in a given state, specifically using Gaussian distributions to generate continuous actions. The policy network of SAC employs the maximum entropy principle, aiming to maximize the reward and the entropy of the policy, thus enhancing exploration capability and improving decision robustness. The objective function of the policy network is defined as:

$$J_\pi(\theta) = \mathbb{E}_{s_t\sim\mathcal{D},a_t\sim\pi_\theta}[\alpha\mathcal{H}(\pi(\cdot\,|s_t)) - Q(s_t, a_t)], \tag{33}$$

here, $\pi_\theta(a_t|s_t)$ is the action distribution generated by the policy network; $Q(s_t, a_t)$ is the current estimated value of the action from the value network; and $\mathcal{H}(\pi(\cdot\,|s_t))$ is the entropy of the policy, measuring the uncertainty of action selection, defined as: $\mathcal{H}(\pi(\cdot\,|s_t)) = -\mathbb{E}_{a_t\sim\pi_\theta}[\log \pi_\theta(a_t|s_t)]$; the parameter α is an entropy regularization coefficient that controls the influence of the entropy term on the objective function.

The action a_t outputted by the policy network is generated from a Gaussian distribution:

$$a_t \sim \pi_\theta(a_t|s_t) = \mathcal{N}(\mu_\theta(s_t), \sigma_\theta(s_t)^2), \tag{34}$$

here, $\mu_\theta(s_t)$ is the mean output of the policy network, and $\sigma_\theta(s_t)$ is the standard deviation outputted by the policy network. The Gaussian distribution can be sampled using the reparameterization trick to reduce the variance of gradient estimates: $a_t = \mu_\theta(s_t) + \sigma_\theta(s_t) \cdot \epsilon, \epsilon \sim \mathcal{N}(0,1)$.

By optimizing the objective function $J_\pi(\theta)$, the policy network is capable of generating action distributions that not only have a high value but also incorporate a certain level of randomness.

(2) **Value Network (Critic Network)**.
The value network assesses the value of the actions produced by the policy through two Q-networks (denoted as Q_1 and Q_2) to reduce overestimation bias. The value network's role is to evaluate the value of the actions generated by the current policy network. SAC introduces a double Q-network

structure to counteract overestimation bias and enhance stability. Specifically, SAC trains two Q-functions Q_1 and Q_2 simultaneously and employs the minimum of the two networks' values as the target value. The goal of the Q-network is to minimize the Bellman error, and its loss function is defined as:

$$J_{\mathrm{Q}}(\phi) = \mathrm{D}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}} \text{úù}Q_{\phi}(s_t, a_t) - y_t)^2 \text{†}, \tag{35}$$

here, $Q_{\phi}(s_t,\ a_t)$ is the current estimate from the Q-network; y_t signifies the target Q-value, defined as:

$$y_t = r_t + \gamma \cdot \mathrm{D}_{a_{t+1}\sim\pi_\theta}\text{Vminù}Q_1(s_{t+1}, a_{t+1}), Q_2(s_{t+1}, a_{t+1})) - \alpha \log \pi_\theta(a_{t+1}|s_{t+1})\text{W}, \tag{36}$$

in this expression, r_t represents the current immediate reward; γ is the discount factor, which measures the importance of future rewards; $\min(Q_1,\ Q_2)$ is used to reduce the overestimation bias of value estimates; and $\alpha\log\ \pi_\theta(a_{t+1}|s_{t+1})$ is the entropy term, which encourages the randomness of the policy. By optimizing the objective function $J_{\mathrm{Q}}(\phi)$, the value network can more accurately assess the value of each action, guiding updates to the policy network.

(3) **Automatic Temperature Adjustment**.

This component dynamically adjusts the entropy coefficient α to maintain a balance between exploration and exploitation. SAC introduces the entropy regularization coefficient α , which encourages the agent to explore more of the unknown state space during the initial phases of training when environmental information is limited. A larger α value enhances exploration capability. Conversely, in the later training stages, as the agent begins to master high-value strategies, a smaller α reduces randomness, heightening the policy's stability and enhancing its exploitation capability, thus achieving a balance between exploration and exploitation. To dynamically adjust α, SAC formulates an adaptive objective function that treats α as a learnable parameter. The optimization objective for automatic temperature adjustment is to minimize the following loss function:

$$J(\alpha) = \mathrm{D}_{a_t\sim\pi_\theta}[-\alpha\log\ \pi_\theta(a_t|s_t) - \alpha \bar{H}\text{¢}], \tag{37}$$

here, $\log\ \pi_\theta(a_t|s_t)$ represents the entropy of the policy, quantifying action randomness, and $\bar{H}$ is a predefined target entropy value that determines the level of randomness within the policy; α is the entropy coefficient to be optimized.

By optimizing $J(\alpha)$, SAC can automatically adjust the entropy coefficient, aligning the policy network's randomness with the exploration needs. Ultimately, the dynamically adjusted α balances the exploration-exploitation relationship. These components enable SAC to exhibit stable, efficient learning capabilities across continuous control tasks.

3.3 Prioritized Experience Replay

Prioritized Experience Replay (PER) is an essential method for enhancing training efficiency in reinforcement learning. The traditional experience replay mechanism samples historical data uniformly, whereas PER introduces a prioritization mechanism to sample high-value experiences, making better use of limited experience data and rendering the learning process more efficient. PER measures the importance of experiences by calculating their Temporal Difference (TD) error, allowing the model to focus more rapidly on significant experiences and speeding up convergence. This can be expressed as:

$$\delta_i = r_i + \gamma Q(s_{i+1}, a_{i+1}) - Q(s_i, a_i). \tag{38}$$

where r_i denotes the immediate reward; $Q(s_i, a_i)$ is the value estimate for the current state and action; $Q(s_{i+1}, a_{i+1})$ is the value estimate for the next state and action; and γ is the discount factor.

The priority of the experience data p_i is defined as:

$$p_i = |\delta_i| + \epsilon, \tag{39}$$

where ϵ is a small constant to avoid zero priorities.

In PER, the sampling probability for each experience is proportional to its priority, thereby increasing the likelihood of sampling experiences with higher priorities. The sampling probability is defined as:

$$P(i) = \frac{p_i^{\alpha}{}_{\alpha,}}{\sum_{\mathrm{j}} p_{\mathrm{j}}} \tag{40}$$

where α controls the effect of priority on sampling. When $\alpha = 0$, PER reduces to uniform sampling; $\sum_{\mathrm{j}} p_{\mathrm{j}}^{\alpha}$ serves as a normalization factor, ensuring that the total sampling probabilities sum to 1.

Furthermore, since non-uniform sampling introduces bias, a weighted correction of the updates for the samples is necessary. The importance-sampling re-weighting mechanism ensures unbiased updates of the model, meaning that high-priority samples do not overly influence the model's learning. The weight is defined as:

$$w_i = \mathrm{k}\frac{1}{N \cdot P(i)}\mathbf{l}^{\beta}, \tag{41}$$

where N is the size of the experience pool; β controls the degree of importance re-weighting and is typically increased from a low value to 1 over the course of training.

3.4 SAC Training Strategy

The SAC algorithm is a reinforcement learning framework based on the principle of maximum entropy, characterized by good stability, high exploration efficiency, and fast convergence speed. In this framework, the rational design of the state space, action space, and reward function is key to achieving effective energy management.

3.4.1 State Space Design

The state space serves as the foundation for the agent's decision-making, necessitating a comprehensive representation of the HEV's operating state and environmental characteristics at the current moment. Based on the core requirements for HEV energy management and the vehicle dynamic model, this study designs the state space s_t as a vector containing the following key variables:

$$\mathbf{s}_t = \mathrm{VSOC}_t, P_{\mathrm{req},t}, v_t, a_t\mathrm{W}, \tag{42}$$

here, SOC_t is the state of charge of the battery, which reflects the current energy level of the battery (typically ranging from 0.2 to 0.8); $P_{req,t}$ is the vehicle's power demand at the current moment, determined by driving conditions and road conditions; v_t represents the vehicle's speed, indicating the current driving state; and a_t is the vehicle's acceleration, further characterizing the vehicle's dynamic performance.

By designing reasonable monitoring of vehicle state parameters, the state space can comprehensively capture the vehicle's dynamic characteristics, energy state, and external environmental information, thereby providing a complete decision-making basis for the SAC algorithm.

3.4.2 Action Space Design

The action space defines the range of decisions available to the agent. In this study, the action space a_t is defined as the engine's output power $P_{eng,t}$, which is the sole variable that the agent decides upon at each time step:

$$\mathbf{a}_t = P_{\text{eng},t}, \tag{43}$$

where $P_{\text{eng},t}$ is the engine output power at the current moment.

In order to ensure the rationality of power distribution, the following constraint conditions must be satisfied:

$$\begin{gathered} P_{\text{req},t} = P_{\text{eng},t} + P_{\text{bat},t}, \\ P_{\text{eng},t} \in [P_{\text{eng,min}}, P_{\text{eng,max}}]. \end{gathered} \tag{44}$$

The power from the battery $P_{bat,t}$ will be automatically calculated from the action variable $P_{eng,t}$ based on the vehicle's power demand as follows:

$$P_{\text{bat},t} = P_{\text{req},t} - P_{\text{eng},t}. \tag{45}$$

Moreover, the constraints regarding battery power and SOC must be adhered to

$$P_{\text{bat},t} \in [P_{\text{bat},min}, P_{\text{bat},max}], \text{SOC}_{min} \leq \text{SOC}_t \leq \text{SOC}_{max}. \tag{46}$$

Through the aforementioned design, the action space for the agent is simplified to controlling the engine's power output. Other variables (such as battery power) are regulated automatically by the dynamic model and constraints, thus simplifying the complexity for both algorithm training and practical application.

3.4.3 Reward Function Design

The design of the reward function directly determines the optimization direction of the SAC algorithm. In the context of HEV energy management, the reward function must take into account fuel consumption, the health status of the battery (changes in State of Health, SOH), and the management objectives regarding battery SOC. This study constructs a reward function of the following form:

$$r_t = -\text{ù} w_1 \cdot \dot{m}_{\text{fuel},t} + w_2 \cdot \Delta\text{SOH}_t + w_3 \cdot |\text{SOC}_t - \text{SOC}_{\text{ref}}|), \tag{47}$$

where $\dot{m}_{fuel,t}$ is the fuel consumption rate at the current moment (in g/s), computed from the engine efficiency model; ΔSOH_t represents the change in the battery's State of Health at the current moment, which is usually related to the depth of discharge (DOD) and charge-discharge rate, calculated as $\Delta SOH_t = k \cdot (DOD_t \cdot C_{rate,t}^2)$, where k is an empirical coefficient, DOD_t represents the current depth of discharge, and $C_{rate,t}$ denotes the current charge-discharge rate; $|SOC_t - SOC_{ref}|$ is the absolute deviation of the battery SOC from its reference value, which penalizes excessive deviations from the target value, with SOC_{ref} set to 0.6 in this study; w_1, w_2, w_3 are weight coefficients used to balance the optimization objectives of fuel consumption, battery health, and SOC management.

By incorporating battery health into the reward function, the optimization process for the EMS simultaneously emphasizes the protection of the battery's health status, mitigating negative impacts on battery life from excessively high rates or depths of discharge. The weight coefficients w_1, w_2, and w_3 can be experimentally tuned based on the significance of different optimization objectives.

The reward function transforms the optimization problem into a minimization problem using negative values, where fuel consumption and the battery's health status directly affect the vehicle's environmental performance and operational efficiency, while the SOC deviation reflects the battery's health management level.

The goal of the SAC algorithm is to maximize the cumulative reward, expressed as:

$$\pi^* = \arg \max_{\pi} \mathrm{D}_{\pi}\, \hat{o}\ddot{e}_{t=0}^{T}\, \gamma^t r_t \pm \tag{48}$$

where π denotes the policy and γ is the discount factor.

4. Simulations and Results Analysis

This section presents simulated experiments based on real driving condition cycles from CLTC-P, designed to validate the effectiveness of the proposed vehicle speed prediction model and the SAC energy management strategy utilizing prioritized experience replay.

4.1 Speed Prediction Parameter Optimization

This section provides a detailed analysis of the impact of the prediction horizon (H) on the performance of the speed prediction model. Setting a larger prediction horizon may improve energy savings to some extent, but it could also affect prediction accuracy and computational complexity. To evaluate the specific influence of the prediction horizon on model performance, we designed five different prediction horizons: no prediction (H=0), H=5, H=10, H=15, and H=20, calculating the fuel consumption and algorithm run time for each case.

The experimental results indicate that as the prediction horizon expands, the overall fuel consumption shows a downward trend. Notably, an optimum balance between fuel consumption and computational time is achieved at H=10, which is therefore regarded as the optimal setting for this study. It is noteworthy that at the H=10 setting, fuel consumption is significantly lower compared to the no prediction condition (H=0), while the computational time does not increase significantly, further validating the effectiveness of the speed prediction method in enhancing energy efficiency.

In addition, Figure 3 illustrates that the Informer-based prediction method employed in this study demonstrates excellent computational efficiency, with run times in the millisecond range. This is primarily attributed to its departure from traditional prediction paradigms based on precise physical modeling. These results not only validate the feasibility and efficiency of the prediction model for real-world applications but also highlight its potential value in speed prediction and energy management.

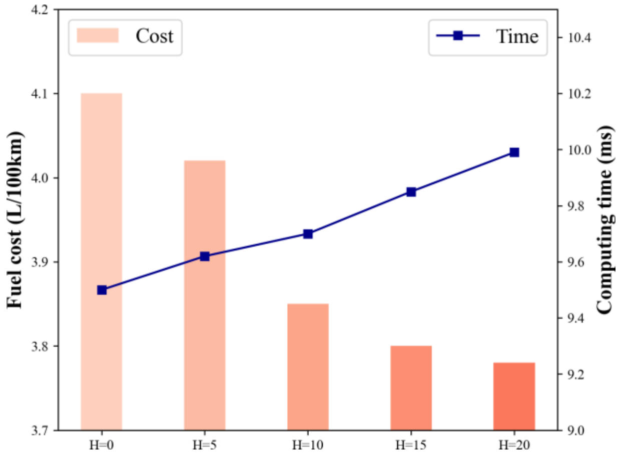

Figure 3 Comparison of Fuel Consumption and Computational Efficiency with Different Prediction Horizons

4.2 Comparison of Speed Prediction Performance

To verify the performance of the Informer model in the task of vehicle speed prediction, several control experiments were designed and comprehensive training and validation tests were conducted using the standard operation cycle dataset CLTC-P. The data from the CLTC-P operation cycle, as illustrated in Figure 4, contains speed profiles under continuous operating conditions, providing rich time series information for the model. The baseline models for comparison include LSTM, GRU, and the original Transformer, which serve to evaluate the performance of each model in predicting vehicle speed. The speed prediction performance metrics utilized are root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), allowing for a comprehensive assessment of the different models' performance in the speed prediction task to facilitate a more thorough and accurate comparison of model effectiveness [32-38].

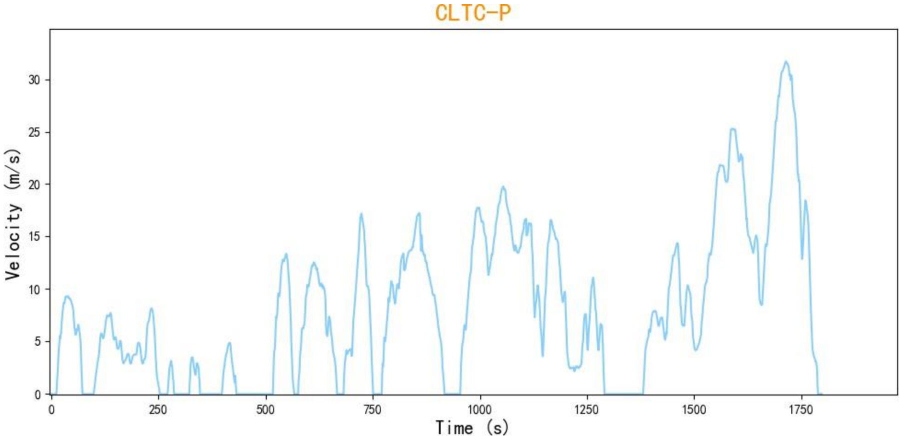

Figure 4 Speed Profile of the CLTC-P Driving Cycle

The results of the control experiments for each baseline speed prediction model are shown in Table 3 and Figure 5. It can be observed that the Informer model significantly outperforms traditional time series models in both prediction accuracy and computational efficiency. Specifically, the Informer improves the RMSE metric by 30% compared to LSTM, demonstrating a marked improvement in error control capability. The MAE is also reduced by 27% with the Informer, showcasing its advantage in minimizing average absolute errors. Furthermore, in terms of the MAPE metric, the Informer achieves a 22% improvement, indicating superb prediction precision.

Table 3: Speed Prediction Performance Metrics for Different Models

Model	RMSE	MAE	MAPE
LSTM	3.00	2.25	9.85
GRU	3.35	2.60	10.32
Transformer	2.50	1.85	8.56
Informer	2.10	1.65	7.62

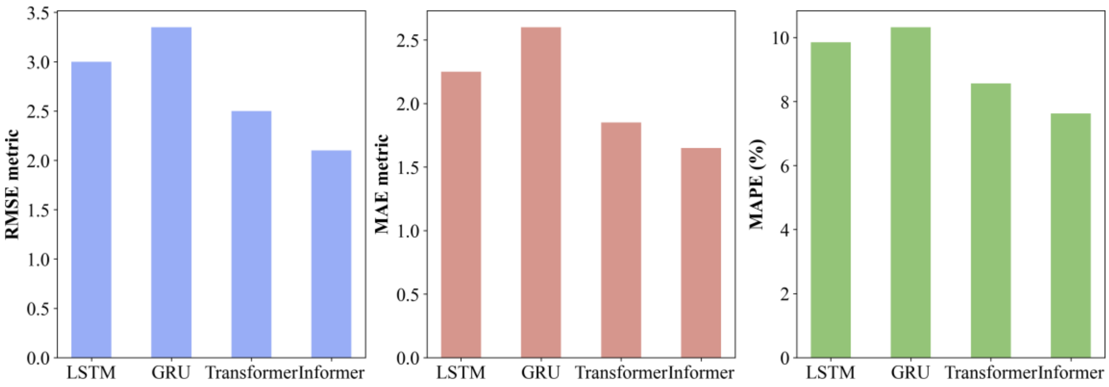

Figure 5 Comparison of Speed Prediction Performance among Different Models

The experimental results indicate that the Informer model can offer higher accuracy and efficiency in speed prediction tasks, surpassing traditional models such as LSTM, GRU, and Transformer, thereby demonstrating its excellent performance and application potential in complex time series prediction. The detailed experiments and significant results affirm the suitability and advantages of using the Informer as a tool for speed prediction in this study.

4.3 Cost Optimization Evaluation for SHEV

In this section, a detailed cost optimization evaluation is conducted regarding the energy management issues for SHEV. During the experiment, the proposed energy management approach based on the Informer vehicle speed prediction and prioritized experience replay, referred to as P-PER-SAC, is compared with traditional SAC and DDPG algorithms.

The comparative experiments are performed under identical simulation environments, establishing four energy management strategies: DDPG, SAC, PER-SAC, and P-PER-SAC. DDPG is a classical deep reinforcement learning algorithm widely applied in continuous control domains. SAC has attracted attention for its exceptional efficiency and robustness; PER-SAC accelerates SAC's convergence by incorporating prioritized experience replay, thus enhancing SAC performance; P-PER-SAC aims to further optimize energy utilization efficiency by integrating the predictive module into PER-SAC. To ensure fair comparisons, all strategies are tested under the same operational conditions. The performance of each strategy is assessed by measuring the vehicle's fuel consumption during the standard testing cycle (CLTC-P), specifically expressed as fuel consumption per kilometer traveled (L/100KM). Each experimental setup is repeated three times to verify the reliability and consistency of results.

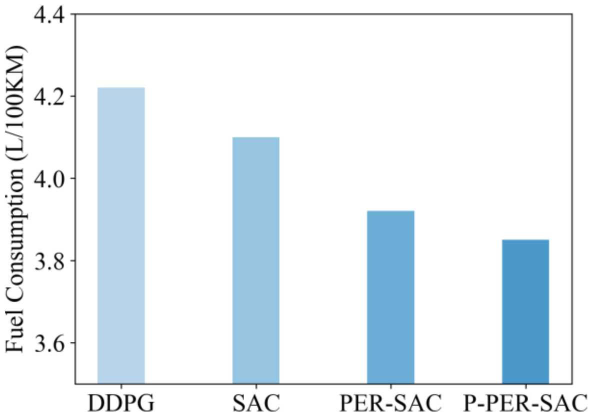

Figure 6 Comparison of Fuel Consumption per 100 Kilometers among Different EMS

The experimental results are presented in Figure 6, showing that DDPG has a fuel consumption of 4.22 L/100KM. SAC improves energy management efficiency by addressing the overestimation problem of DDPG, further reducing fuel consumption to 4.10 L/100KM. Notably, the approach proposed in this study, incorporating prioritized experience replay and a vehicle speed prediction module, significantly enhances SAC's performance. Specifically, the integration of prioritized experience replay reduces fuel consumption to 3.92 L/100KM, and the addition of the speed prediction module in the P-PER-SAC strategy achieves optimal performance with a fuel consumption of 3.85 L/100KM. Compared to DDPG and SAC, P-PER-SAC clearly reduces energy consumption, highlighting the critical role of integrating prioritized experience replay and speed prediction in enhancing energy optimization efficiency. This also validates the rationale behind incorporating prioritized experience replay and speed prediction into the SAC-based energy management framework proposed in this study.

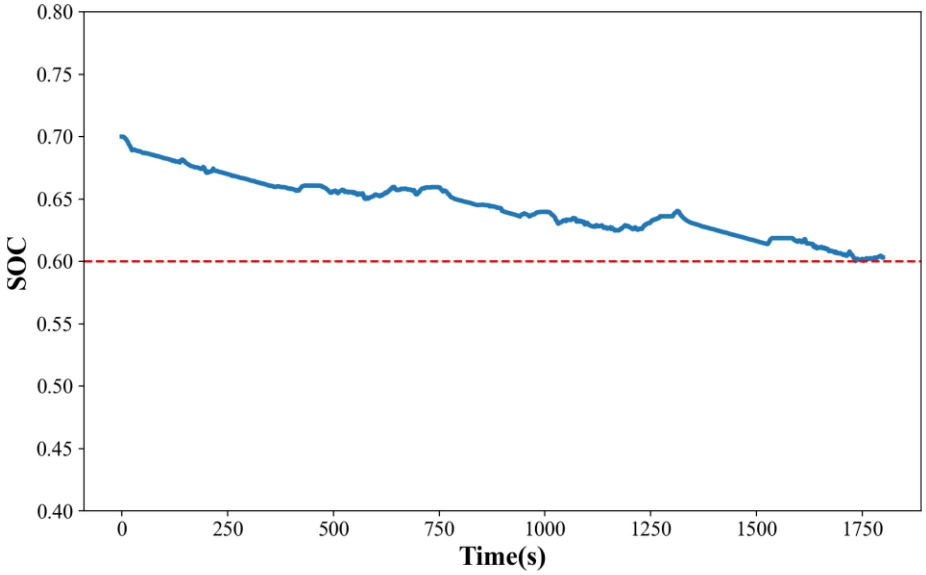

Figure 7 SOC Variation Curve of P-PER-SAC

Moreover, to validate the effectiveness of the proposed method in maintaining charge, Figure 7 illustrates the SOC variation curve of the approach under the CLTC-P driving cycle. The initial SOC for the simulation experiment is set at 0.7; after a period of operation, the energy management strategy consistently maintains the SOC around 0.6. At this state, the efficiency of battery charging and discharging is relatively high, aligning with the design objectives of this study and helping to maximize battery utilization efficiency. Additionally, it is observed from the figure that the amplitude of SOC variation remains relatively small over time, indicating the effectiveness of the method in optimizing energy management. Maintaining a high SOC level can reduce energy losses and extend the battery's service life. Therefore, the proposed method demonstrates significant advantages in terms of battery SOC stability, validating its effectiveness and practicality in real-world applications.

Finally, Figure 8 shows the changes in rewards during the training process of the proposed method, where the red line represents the smoothed reward values and the yellow line represents the raw reward values. It is evident from the figure that as training progresses, especially when the episode exceeds 200, rewards tend to stabilize, indicating that the method has converged during the training process. The stability of reward values signifies that the model has reached a relatively optimal strategy after prolonged training, further illustrating the effectiveness and stability of the method in SHEV energy management.

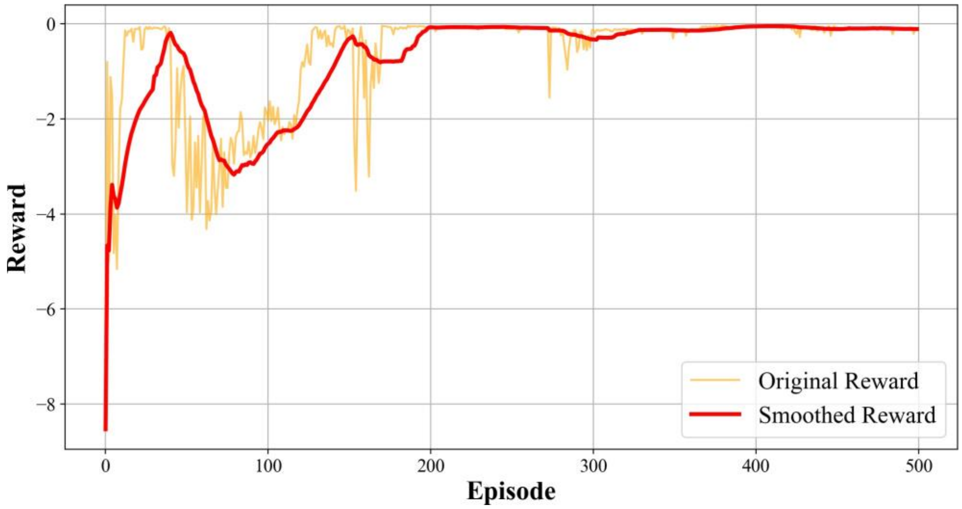

Figure 8 Changes in Rewards During the Training Process

In summary, the proposed method exhibits significant advantages in energy management compared to traditional DDPG and SAC algorithms. By incorporating prioritized experience replay and the vehicle speed prediction module, the method not only improves fuel efficiency but also effectively maintains the battery SOC in an optimal state. This enables the battery to operate efficiently while reducing energy losses and extending its lifespan. These results indicate that the proposed method possesses substantial application potential and practical feasibility in optimizing energy management for hybrid electric vehicles, offering new insights and directions for future research and real-world applications.

5. Conclusions

This paper conducts an in-depth study on the energy management issues for SHEV and proposes a SAC energy management method based on Informer vehicle speed prediction and prioritized experience replay. By comparing this method with traditional algorithms such as DDPG, SAC, and PER-SAC within the same simulation environment, we evaluate various energy management strategies based on two key indicators: fuel consumption and battery SOC stability. The proposed method achieves a fuel consumption reduction to 3.85 L/100KM while maintaining the battery SOC stable around the preset value of 0.6, representing a 6.1% improvement over the traditional SAC-based energy management system. This demonstrates the crucial role of prioritized experience replay and the speed prediction module in optimizing the energy management efficiency of the SAC algorithm. Additionally, the analysis of the reward variations throughout the training process indicates that the rewards stabilize as training progresses, further confirming the convergence and robustness of the proposed method. In summary, the energy management strategy proposed in this study exhibits significant advantages over traditional algorithms, performing excellently in optimizing fuel consumption and maintaining battery SOC stability. It holds strong application potential and practical feasibility [39-44]. This research outcome offers new insights and directions for the future development of energy management strategies for hybrid electric vehicles, with the potential to yield significant energy-saving benefits in practical applications.

Funding

This research has been partly funded by the National Natural Science Foundation China (NSFC) through award No. 51976143. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSFC.

Author Contributions

Weidong Huang contributed to the development of the predictive energy management strategy, implementation of the hybrid electric vehicle model, and analysis of simulation results. Jiahuai Ma supervised the research, provided guidance on reinforcement learning methodologies, and contributed to manuscript revisions.

Institutional Reviewer Board Statement

Not applicable

Informed Consent Statement

Not applicable

Data Availability Statement

The data supporting the findings of this study are available from the corresponding author upon request.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] S.Zhou, Z.Chen, D.Huang, and T.Lin, Model prediction and rule based energy management strategy for a plug-in hybrid electric vehicle with hybrid energy storage system, IEEE Trans. Power Electron., vol.36, no.5.pp. 5926–5940, May 2021.

[2] H. Tian, X. Wang, Z. Lu, Y. Huang, and G. Tian, “Adaptive fuzzy logic energy management strategy based on reasonable SOC reference curve for online control of plug - in hybrid electric city bus,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 5, pp. 1607 - 1617, May 2018.

[3] J. Liu, Y. Chen, J. Zhan, and F. Shang, “Heuristic dynamic programming based online energy management strategy for plug - in hybrid electric vehicles,” IEEE Trans. Veh. Technol., vol. 68, no. 5, pp. 4479 - 4493, May 2019.

[4] C. Hou, M. Ouyang, L. Xu, and H. Wang, “Approximate Pontryagin’s minimum principle applied to the energy management of plug - in hybrid electric vehicles,” Appl. Energy, vol. 115, pp. 174 - 189, 2014.

[5] Z. Chen, C. C. Mi, R. Xiong, J. Xu, and C. You, “Energy management of a power - split plug - in hybrid electric vehicle based on genetic algorithm and quadratic programming,” J. Power Sources, vol. 248, pp. 416 - 426, Feb. 2014.

[6] Li Y, Tao J, Xie L, Zhang R, Ma L, Qiao Z. Enhanced Q-learning for real-time hybrid electric vehicle energy management with deterministic rule[J]. Measurement and Control. 2020, 53(7-8):1493-1503.

[7] A. M. Phillips, M. Jankovic and K. E. Bailey, Vehicle system controller design for a hybrid electric vehicle[C]. IEEE International Conference on Control Applications. Conference Proceedings , Anchorage, AK, USA, 2000, pp. 297-302.

[8] Lv Ming, Yang Ying, Lijie Liang, et al. Energy Management Strategy of a Plug-in Parallel Hybrid Electric Vehicle Using Fuzzy Control[J]. Energy Procedia, 2017, 105.

[9] Gao C, Zhao J, Wu J. Optimal fuzzy logic based energy management strategy of battery/supercapacitor hybrid energy storage system for electric Vehicles[C].2016 12th World Congress on Intelligent Control and Automation, Guilin, China, June, 2016.

[10] Tang, X., Chu, L., Xu, N.et al. Energy Management of Planetary Gear Hybrid Electric Vehicle Based on Improved Dynamic Programming[C]. Neural Information Processing. ICONIP 2017.

[11] Shi, D., Guo, J., Liu, K., et al. Research on an Improved Rule-Based Energy Management Strategy Enlightened by the DP Optimization Results[J]. Sustainability 2023, 15.

[12] Y. Farajpour, H. Chaoui, M. Khayamy, S. Kelouwani and M. Alzayed, A Hybrid Energy Management Strategy Based on ANN and GA Optimization for Electric Vehicles[J]. 2022 IEEE Vehicle Power and Propulsion Conference (VPPC), Merced, CA, USA, 2022, pp. 1-6.

[13] Yapeng Li, Xiaolin Tang, Xianke Lin. The role and application of convex modeling and optimization in electrified vehicles[J]. Renewable and Sustainable Energy Reviews, Volume 153, 2022.

[14] Zhang Y, Chu L, Fu Z, et al. An improved adaptive equivalent consumption minimization strategy for parallel plug-in hybrid electric vehicle[C]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2019, 233(6): 1649 -1663.

[15] Zhang F, Hu X, Langari R, et al. Energy management strategies of connected HEVs and PHEVs: recent progress and outlook[J]. Prog Energy Combust Sci, 2019, 73:235-56.

[16] Zhou et al. Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle[J]. Applied Energy, 2019，25.

[17] Liu T, Zou Y, Liu D, et al. Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle[J]. IEEE Trans Ind Electron, 2015, 62(12):7837-46.

[18] Xiong R, Cao J, Yu Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle[J]. Applied Energy, 2018, 211:538-48.

[19] Wu J, He H, Peng J, et al. Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus[J]. Applied Energy, 2018, 222: 799-811.

[20] Da Wang, Lei Mei, Chuanxue Song, et al. Energy management strategy with mutation protection for fuel cell electric vehicles[J]. International Journal of Hydrogen Energy, 2024, 63:48-58.

[21] Tan H, Zhang H, Peng J, et al. Energy management of hybrid electric bus based on deep reinforcement learning in continuous state and action space[J]. Energy Convers Manag, 2019;195:548-60.

[22] Wu Y, Tan H, Peng J, et al. Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series parallel plug-in hybrid electric bus[J]. Applied Energy 2019, 247:454-66.

[23] Sun, X., et al. "A Markov chain-based velocity prediction model for hybrid electric vehicle energy management." Journal of Automotive Engineering, 2018.

[24] Zhang, H., et al. "ARIMA-based velocity prediction and its application in hybrid electric vehicle energy management." Energy Conversion and Management, 2020.

[25] Liu, Y., et al. "Support vector machine-based velocity prediction for hybrid electric vehicles." Transportation Research Part D, 2019.

[26] Chen, Z., et al. "Real-time velocity prediction using LSTM for hybrid electric vehicle control." IEEE Transactions on Intelligent Vehicles, 2021.

[27] Wang, Y., et al. "Vehicle-to-infrastructure velocity prediction for hybrid electric vehicle energy management." Transportation Research Part C, 2017.

[28] Ding, J., et al. "Random forest-based velocity prediction using connected vehicle data." Applied Energy, 2019.

[29] Wang, S., et al. "Integrated LSTM prediction and DRL optimization for hybrid electric vehicle energy management." IEEE Transactions on Vehicular Technology, 2022.

[30] Zou, X., et al. "Fuzzy logic-based predictive energy management for hybrid vehicles." Expert Systems with Applications, 2020.

[31] Ebbesen S, Elbert P, Guzzella L. Battery state-of-health perceptive energy management for hybrid electric vehicles[J]. IEEE Transactions on Vehicular technology, 2012, 61(7): 2893-2900.

[32] Z. Luo, H. Yan, and X. Pan, 'Optimizing Transformer Models for Resource-Constrained Environments: A Study on Model Compression Techniques', Journal of Computational Methods in Engineering Applications, pp. 1–12, Nov. 2023, doi: 10.62836/jcmea.v3i1.030107.

[33] H. Yan and D. Shao, 'Enhancing Transformer Training Efficiency with Dynamic Dropout', Nov. 05, 2024, arXiv: arXiv:2411.03236. doi: 10.48550/arXiv.2411.03236.

[34] Y. Liu and J. Wang, 'AI-Driven Health Advice: Evaluating the Potential of Large Language Models as Health Assistants', Journal of Computational Methods in Engineering Applications, pp. 1–7, Nov. 2023, doi: 10.62836/jcmea.v3i1.030106.

[35] Y. Gan and D. Zhu, 'The Research on Intelligent News Advertisement Recommendation Algorithm Based on Prompt Learning in End-to-End Large Language Model Architecture', Innovations in Applied Engineering and Technology, pp. 1–19, 2024.

[36] D. Zhu, Y. Gan, and X. Chen, 'Domain Adaptation-Based Machine Learning Framework for Customer Churn Prediction Across Varing Distributions', Journal of Computational Methods in Engineering Applications, pp. 1–14, 2021.

[37] H. Zhang, D. Zhu, Y. Gan, and S. Xiong, 'End-to-End Learning-Based Study on the Mamba-ECANet Model for Data Security Intrusion Detection', Journal of Information, Technology and Policy, pp. 1–17, 2024.

[38] D. Zhu, X. Chen, and Y. Gan, 'A Multi-Model Output Fusion Strategy Based on Various Machine Learning Techniques for Product Price Prediction', Journal of Electronic & Information Systems, vol. 4, no. 1.

[39] P. Ren and Z. Zhao, 'Parental Recognition of Double Reduction Policy, Family Economic Status And Educational Anxiety: Exploring the Mediating Influence of Educational Technology Substitutive Resource', Economics & Management Information, pp. 1–12, 2024.

[40] Z. Zhao, P. Ren, and Q. Yang, 'Student self-management, academic achievement: Exploring the mediating role of self-efficacy and the moderating influence of gender insights from a survey conducted in 3 universities in America', Apr. 17, 2024, arXiv: arXiv:2404.11029. doi: 10.48550/arXiv.2404.11029.

[41] P. Ren, Z. Zhao, and Q. Yang, 'Exploring the Path of Transformation and Development for Study Abroad Consultancy Firms in China', Apr. 17, 2024, arXiv: arXiv:2404.11034. doi: 10.48550/arXiv.2404.11034.

[42] Z. Zhao, P. Ren, and M. Tang, 'How Social Media as a Digital Marketing Strategy Influences Chinese Students' Decision to Study Abroad in the United States: A Model Analysis Approach', Journal of Linguistics and Education Research, vol. 6, no. 1, pp. 12–23, 2024.

[43] Z. Zhao, P. Ren, and M. Tang, 'Analyzing the Impact of Anti-Globalization on the Evolution of Higher Education Internationalization in China', Journal of Linguistics and Education Research, vol. 5, no. 2, pp. 15–31, 2022.

[44] M. Tang, P. Ren, and Z. Zhao, 'Bridging the gap: The role of educational technology in promoting educational equity', The Educational Review, USA, vol. 8, no. 8, pp. 1077–1086, 2024.