machines

MDPI

Article

An Energy-Efficient Driving Method for Connected and Automated Vehicles Based on Reinforcement Learning

Haitao Min [1], Xiaoyong Xiong [1], Fang Yang [2], Weiyi Sun [1], Yuanbin Yu [1] and Pengyu Wang [1,*]

[1] State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130012, China
[2] General Research and Development Institute, China FAW Corporation Limited, Changchun 130013, China
* Correspondence: wangpy@jlu.edu.cn

Abstract: The development of connected and automated vehicles (CAV) technology not only helps to reduce traffic accidents and improve traffic efficiency, but also has significant potential for energy saving and emission reduction. Using the dynamic traffic flow information around the vehicle to optimize the vehicle trajectory is conducive to improving the energy efficiency of the vehicle. Therefore, an energy-efficient driving method for CAVs based on reinforcement learning is proposed in this paper. Firstly, a set of vehicle trajectory prediction models based on long and short-term memory (LSTM) neural networks are developed, which integrate driving intention prediction and lane change time prediction to improve the prediction accuracy of surrounding vehicle trajectories. Secondly, an energy-efficient driving model is built based on Proximity Policy Optimization (PPO) reinforcement learning. The model takes the current states and predicted trajectories of surrounding vehicles as input information, and outputs energy-saving control variables while taking into account various constraints, such as safety, comfort, and travel efficiency. Finally, the method is tested by simulation on the NGSIM dataset, and the results show that the proposed method can save energy consumption by 9–22%.

Keywords: connected and automated vehicles; energy-efficient driving; reinforcement learning; long short-term memory; proximal policy optimization

Citation: Min, H.; Xiong, X.; Yang, F.; Sun, W.; Yu, Y.; Wang, P. An Energy-Efficient Driving Method for Connected and Automated Vehicles Based on Reinforcement Learning. *Machines* **2023**, *11*, 168. https://doi.org/10.3390/machines11020168

Academic Editor: Nasser L. Azad

Received: 11 December 2022
Revised: 11 January 2023
Accepted: 23 January 2023
Published: 26 January 2023

1. Introduction

In recent years, the technology of connected and automated vehicles (CAV) has developed rapidly. In recent years, the technology of the Internet of Vehicles and autonomous vehicles (CAV) has developed rapidly. CAV can obtain massive information through networks and sensors, then complete information processing and trajectory planning through high-performance computing platforms and artificial intelligence methods, and, finally, accurately control vehicle movement; therefore, CAV are expected to solve the problems of traffic congestion and road safety [1]. Safety, comfort, and efficient driving have always been the focus of research in the field of CAV technology. It is worth noting that these features of CAV are also of a significant aid to energy conservation and emission reduction. How to make full use of the CAV's network information and autonomous driving capabilities to achieve eco-driving has become a hot topic for researchers [2].

The energy efficiency of vehicles is related to various factors, such as driving conditions, drivetrain efficiency, and energy management strategies. Driving behavior can change the driving conditions, thus affecting the energy consumption of vehicles [3,4]. Some studies show that changing a driver's driving style will affect fuel consumption by 5–20% [5], and an aggressive driving style will increase the energy consumption of pure electric vehicles by 7% [3]. Therefore, many researchers have sought to improve vehicle energy efficiency by changing drivers' driving behaviors, such as improving drivers' driving habits through audio and visual alerts [6,7], providing feedback on drivers' driving behavior through warning, scoring and ranking [8], and developing optimal speed

suggestion systems [9]. However, in practice some drivers may not follow the expected behavior of these methods, thus the effectiveness of these methods will be severely reduced. In contrast, the driving behavior of CAV is completed automatically, and the planned energy-saving trajectory can always be tracked more precisely, allowing for more stable energy-saving effects.

Most of the early research on energy-saving cruise control focused on the optimal control of vehicle velocity. The pulse and glide (PnG) strategy is a typical strategy to save energy by optimizing the speed control of vehicle. With this strategy, the vehicle is accelerated to high speed during the pulse phase using an acceleration in the high-efficiency range of the engine, and glides to low speed during the glide phase with the engine off [10]. PnG strategy has been shown to be effective for fuel vehicles [10], electric vehicles [11], and hybrid vehicles [12], and has been shown to save more than 20% of energy consumption. However, the PnG strategy ignores the comfort constraint and also causes a great disturbance to the overall traffic flow, which is of poor practicality [13].

Therefore, energy-efficient driving strategies must take into account a variety of factors, such as comfort, safety, and dynamic traffic flow while improving energy efficiency. CAV's ability to obtain anticipatory information is helpful in this regard. Studies have shown that fuel vehicles with as little as 7 s of traffic foresight can produce the same energy savings as hybrid vehicles [14]. Forward-looking information, such as vehicle speed limits, gradients, and traffic signals on the preceding roads, can be obtained from networked navigation maps or predicted from offline data [15,16]. Anticipating this information allows the vehicle to optimize speed planning, adjust gears and avoid unnecessary braking, thus reducing energy consumption [17–20].

In addition to predicting static information, the prediction of dynamic traffic information is also important for energy-efficient driving. Predicting the surrounding vehicle motion during trajectory planning is helpful to meet the safety constraint and also avoid excessive speed fluctuations, thus improving the energy-saving effect as well as the feasibility of the trajectory [21,22]. Constant velocity, constant acceleration or constant rate of change of acceleration can be used as simple velocity prediction models to improve energy efficiency [23,24]. As the prediction time increases, it is difficult for the vehicle to maintain a constant speed or acceleration, hence the prediction error increases.

For medium- and long-term trajectory prediction, the data-driven models represented by recurrent neural network (RNN) and long short-term memory (LSTM) neural network have good performance. Some researchers have built vehicle trajectory prediction models based on LSTM, such as LSTM prediction model using encoder–decoder framework [25,26], LSTM prediction model using social pooling mechanism [27,28], LSTM prediction model incorporating spatio-temporal characteristics [29,30], etc. All these methods can show higher trajectory prediction accuracy than constant velocity and constant acceleration models in a longer prediction time horizon. However, these models are rarely used in eco-cruising because most of the current methods are based on optimization methods, such as MPC or optimal control [31,32], which would become difficult to solve if such data-driven models are embedded.

In general, current energy-saving driving methods still have some limitations, which are mainly reflected in the following two aspects. Firstly, the trajectory prediction for surrounding vehicles is relatively simple and cannot accurately predict the vehicle movement in the medium and long term, thus making it difficult to handle dynamic and complex traffic flow scenarios. Secondly, existing studies usually improve vehicle energy efficiency through optimal control of speed, ignoring the role of lane changes. Some studies that have taken lane selection into account only treat lane change as a transient state transition, ignoring the details of the lateral motion of the vehicle [33–35], thus ignoring the safety, kinematic, and other constraints of the vehicle in lateral motion.

In order to solve the above problems, an energy-efficient driving method based on reinforcement learning (RL) is proposed in this work. A set of trajectory prediction models based on LSTM are established to improve the accuracy of medium- and long-term tra-

jectory prediction of surrounding vehicles. Then, combined with the trajectory prediction information, an energy-saving driving model that takes into account multiple constraints is developed. The model is optimized by the Proximal Policy Optimization (PPO) reinforcement learning to obtain the optimal parameters. The main contributions of this work are as follows:

1. A set of vehicle trajectory prediction models based on hierarchical LSTM is constructed. Accurate trajectory prediction is conducive to improving the adaptability of energy-saving driving methods under dynamic traffic flow. The LSTM-based method has a low error in the medium- and long-term trajectory prediction. Considering that it is difficult to fit trajectories with different driving intentions by a single LSTM model, the trajectory prediction problem is decomposed into three steps: driving intention prediction, lane change time prediction and trajectory prediction, thus reducing the fitting difficulty and improving the prediction accuracy.
2. A deep reinforcement learning-based method for energy-efficient driving under multiple constraints is proposed. The method is no longer limited to the speed planning of a single lane, but takes into account both the longitudinal and lateral motion of the ego vehicle. In addition, the method takes into account various constraints, such as safety, comfort, and travel efficiency under dynamic traffic while improving energy efficiency.

The rest of this paper is organized as follows: The overall framework of the proposed method is given in Section 2. Section 3 presents the proposed hierarchical LSTM trajectory prediction model. Section 4 describes the proposed deep RL-based energy-efficient driving method. Simulation results and discussions are reported in Section 5. Finally, the conclusions are drawn in Section 6.

2. Architecture of the Energy-Efficient Driving Approach

As shown in Figure 1, the ego vehicle obtains the position, speed, acceleration and other states information of surrounding vehicles through V2V and sensors, such as LIDAR and cameras. Then the trajectory prediction system takes this information as an input and outputs the predicted trajectories of surrounding vehicles through three steps: driving intention prediction, lane change time prediction, and trajectory prediction. Combining the ego vehicle states, environmental information, and the prediction information, the energy-saving control system calculates the energy-optimal control variables under multiple constraints. The underlying execution system controls the motion of the vehicle according to the given control variables.

The surrounding vehicle trajectory prediction system consists of several LSTM neural network models, and the model parameters are obtained by training a large amount of trajectory data. The energy saving control system consists of a multilayer fully connected neural network, and the network parameters are trained by a PPO reinforcement learning method. The two systems receive information from the perception system as input and output control variables to the execution system to achieve energy efficiency improvement.

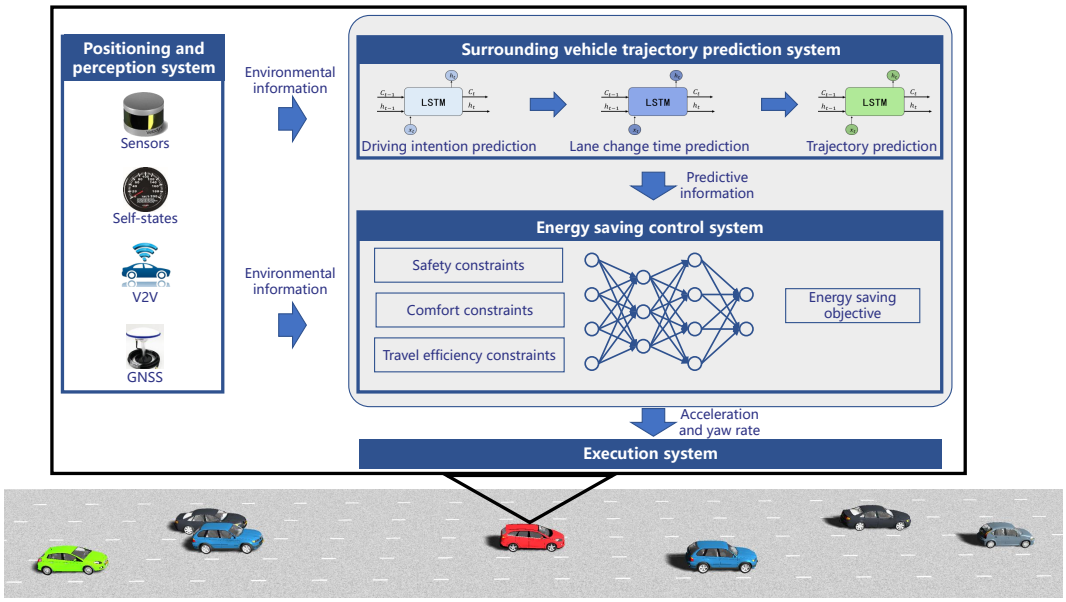

Figure 1. The implementation architecture of the proposed method.

3. Vehicle Trajectory Prediction

In this paper, a hierarchical LSTM-based model is used to predict the trajectory of surrounding vehicles, and the trajectory prediction problem is decomposed into three steps: driving intention prediction, lane change time prediction, and trajectory prediction. The development of these models consists of three main tasks: first, configuring the trajectory dataset for each model, then constructing and training each model, and, finally, logically combining these models.

3.1. Vehicle Trajectory Dataset Configuration

The Next Generation Simulation (NGSIM) I-80 dataset [36] is used for model training, The dataset was collected on the highway and contains more than 5000 vehicle trajectories with more than 3,000,000 trajectory points. Each trajectory acquisition frequency is 10 Hz, i.e., 10 frames per second. Data with conflicting trajectories and abrupt position changes in the raw data are removed. Then the entire dataset is partitioned, as shown in Figure 2. The total dataset is first divided into left and right lane change and lane keeping datasets, and the left and right lane change datasets are then divided into datasets with different lane change times, for a total of 3 layers, forming 14 different datasets.

For each sample in all datasets, the previous 50 frames of the current frame are used as the input observations. The observed data features contained in each frame are shown in Formula (1).

$$obs_i = \{x_i, y_i, vx_i, vy_i, ax_i, ay_i, lane_i, LH_i, CH_i, RH_i, dtoL_i\} \quad (1)$$

where x and y are the longitudinal and lateral location coordinates; vx and vy are the longitudinal and lateral velocities; ax and ay are the longitudinal and lateral accelerations; $lane$ is the lane where the vehicle is located; LH, CH, and RH are the headway of the nearest vehicle in front of the left lane, current lane, and right lane, respectively; and $dtoL$ is the lateral distance between the vehicle and the centerline of the lane.

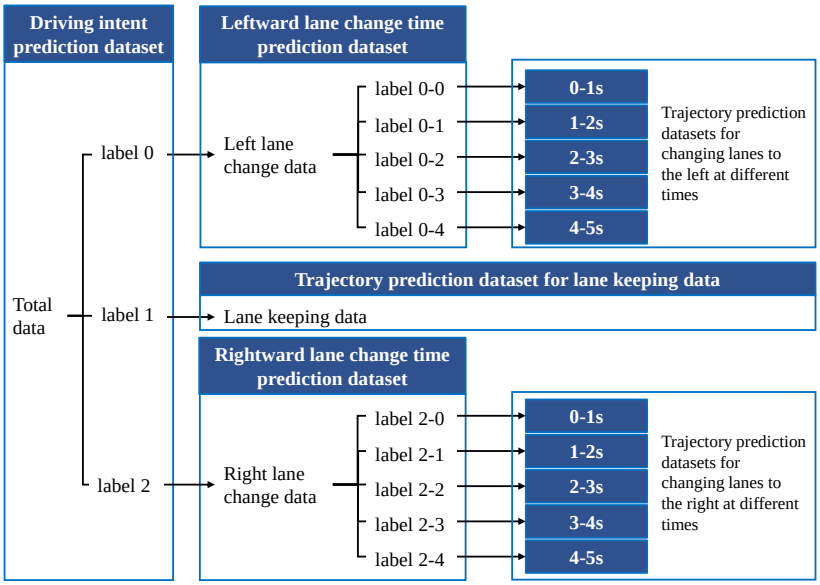

Figure 2. Dataset segmentation scheme.

The target outputs of the models are different for different steps. The target output of the driving intention prediction model is one of three different driving intentions: left lane changing, right lane changing, or lane keeping. The target output of the lane change time prediction model is the time range in which the lane change occurs, including 0–1 s, 1–2 s, ..., 4–5 s. The trajectory prediction models are expected to output the predicted values of vehicle coordinates for the next 50 frames.

3.2. Prediction Model Construction and Training

The 14 separate datasets are used to train the models for different steps. As shown in Figure 3, all models are composed of two layers of LSTM and one layer of fully connected dense network (DENSE). LSTM is an improvement on RNN with better performance for solving long-term dependency problems, and its principle is described in detail in the literature [37]. The input to all models is a sequence of observations of dimension 11 and length 50. The driving intention prediction model and the lane change time prediction models are essentially classification models, of which the output results are often encoded by One-Hot, i.e., each category is treated as a distinct dimension of the feature space [38]. There are three possible outputs of the intention prediction model, so the dimension of the predicted value $\hat{y}_i$ is 3. Similarly, the outputs of the lane change time prediction model have 5 dimensions in One-Hot encoding. The cross-entropy function, which is commonly used by classification models [39], is used as the loss function for the driving intention prediction model and the lane change time prediction models. The trajectory prediction models are essentially regression models, and the outputs are prediction sequences of length 50. The loss functions of trajectory prediction models use the mean square error between the predicted and true values, which is commonly used by regression models.

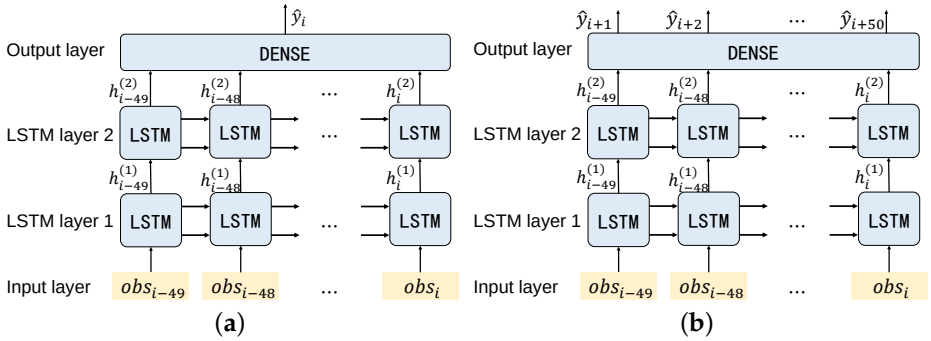

Figure 3. LSTM models structure: (**a**) the driving intention and lane change time prediction models. (**b**) The vehicle trajectories prediction models.

The models are trained using the above configurations. The convergence processes of the loss functions of the driving intention prediction model and the lane change time prediction models are shown in Figure 4. The three models achieved accuracy rates of 98.2%, 91.9%, and 90.8% in their respective test datasets.

According to the dataset division in Figure 2, different trajectory prediction models are trained separately. The root mean square errors (RMSEs) of the predicted trajectories of each model are shown in Table 1. LX-1 and LY-1 denote the lateral and longitudinal trajectory prediction models for left lane change occurring in 0–1 s, RX-1 and RY-1 denote the lateral and longitudinal trajectory prediction models for right lane change occurring in 0–1 s, and CX and CY denote the lateral and longitudinal trajectory prediction models under lane keeping intention, respectively. The errors shown in the table are all relatively low, which is due to the lower fitting difficulty of the sub-datasets after the dataset segmentation. However, these are not the final errors of the whole set of models. In addition to the prediction accuracy of each model in the sub-datasets, the final errors are related to the accuracy of both the driving behavior prediction model and the lane change time prediction models.

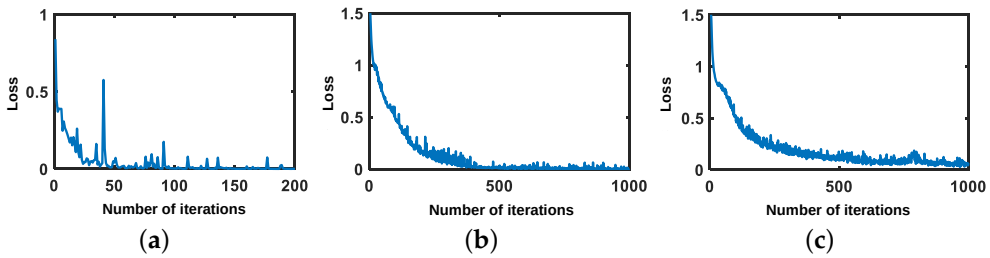

Figure 4. The convergence processes of the loss functions: (**a**) the driving intention prediction model. (**b**) Leftward lane change time prediction model. (**c**) Rightward lane change time prediction model.

Table 1. The RMSEs of the predicted trajectories of each trajectory prediction model.

Model	**LX-1**	**LX-2**	**LX-3**	**LX-4**	**LX-5**	**CX**	**RX-1**	**RX-2**	**RX-3**	**RX-4**	**RX-5**
RMSE	0.067	0.109	0.124	0.116	0.121	0.119	0.174	0.096	0.081	0.112	0.166
Model	**LY-1**	**LY-2**	**LY-3**	**LY-4**	**LY-5**	**CY**	**RY-1**	**RY-2**	**RY-3**	**RY-4**	**RY-5**
RMSE	0.893	1.041	1.083	1.546	1.168	0.521	2.901	0.951	1.422	0.996	1.597

3.3. Model Combinations

As shown in Figure 5, in the first step, the driving intention prediction model is invoked. If the intention prediction result is left lane change or right lane change, the second step is performed, i.e., the corresponding lane change time prediction model is invoked. Then the third step is performed, which invokes the corresponding trajectory prediction model to output the final prediction of the trajectory. If the intention prediction

result is lane keeping, the trajectory prediction model corresponding to the lane keeping intention is invoked.

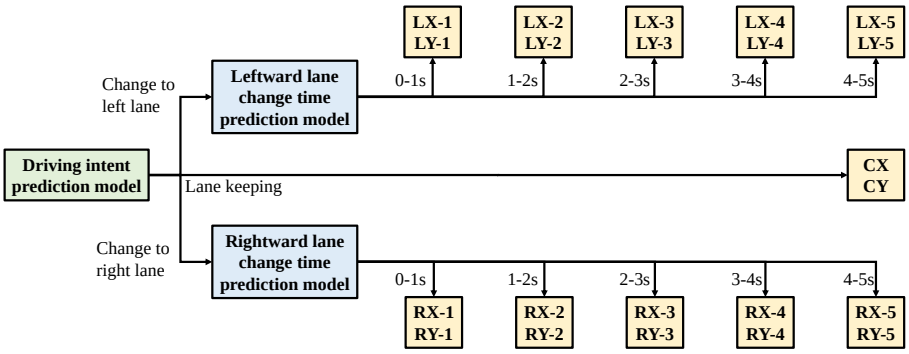

Figure 5. The logic of model invocation for each step.

The proposed method (Hierarchical LSTM) is compared with the polynomial, constant acceleration, and integral LSTM models using the NGSIM dataset, and the results are shown in Figure 6 and Table 2. Figure 6 shows the prediction results for a single trajectory. It can be seen that the prediction trajectory based on polynomial and constant acceleration is smooth, however, the error gradually increases as the prediction time horizon is lengthened, while the proposed hierarchical LSTM has higher prediction accuracy in the case of longer time horizon.

The method is evaluated by RMSE, average displacement error (ADE), and final displacement error (FDE) metrics. Table 2 shows that the constant acceleration as well as the polynomial prediction error is larger in the long-term prediction of 5 s, while the integral LSTM prediction error is reduced, and the error of the hierarchical LSTM method proposed in this paper is the lowest.

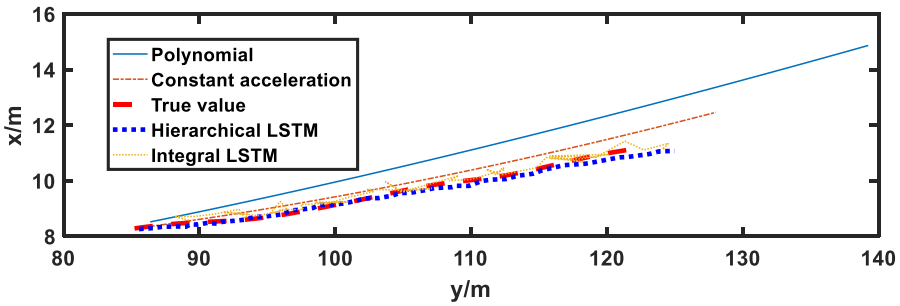

Figure 6. Comparison of prediction results of different methods.

Table 2. Comparison of RMSE, ADE, and FDE of different methods over a 5-second prediction horizon.

Method		Polynomial	Constant Acceleration	Integral LSTM	Hierarchical LSTM
RMSE/m	x	1.51	1.42	0.53	0.34
	y	10.09	9.34	4.42	2.94
ADE/m		5.61	5.27	2.75	2.36
FDE/m		12.24	11.43	6.53	5.42

4. Energy Efficient Driving Model with Multiple Constraints

Reinforcement learning (RL) is a machine learning approach through the interaction of an agent with an external environment. The agent employs a policy to interact with the external environment, collects experience during the interaction, and uses the experience to improve the policy for maximum reward [40]. The behavioral logic of the agent is called the policy function $p_\theta(a|s)$, which makes an action a based on the observed state s. The policy

function is a mapping of states s to actions a, which can be either a deterministic mapping or a probability distribution of actions. θ is the parameter of the policy, and in deep RL, the policy function is generally in the form of a neural network, where θ are the parameters of the policy neural network. PPO reinforcement learning is a policy learning method that is a improvement of policy gradient (PG) [41]. The detailed principle of the PPO algorithm is described in [42], and we will describe the process of building and training the network for the energy-efficient driving problem.

4.1. PPO Reinforcement Learning Model Building

We use a RL-based approach to build and train an energy-efficient driving model based on the principles of the PPO algorithm. The method is mainly divided into two parts: empirical data collection and network parameter optimization, where the model architecture of the empirical data collection part is shown in Figure 7. The specific steps of the empirical data collection are shown below:

1. Initialize the network parameters and the environment at the start moment. Assign random initial parameters to the value network and the policy network, and set the environment to the initial state.
2. The value network accepts the state s_t of the environment as input and outputs an estimate val_t of the value of the state s_t. The policy network accepts the state s_t of the environment as input and outputs the sampling probability $p(a_t|s_t)$ of each action in state s_t.
3. Sample each action according to the probability $p(a_t|s_t)$, output the resulting action a_t, and calculate the logarithm of the action probability density $logProb_t$.
4. Apply the obtained action a_t to the environment. The environment updates the state in response to a_t and returns the reward value r_t, the updated state s_{t+1}, and a flag $done_t$ for whether the episode is completed or not. If the episode is completed, use step 1 for initialization.
5. Record $s_t, a_t, r_t, val_t, logProb_t, done_t$ from the above steps in the experience pool and repeat the above steps until the experience pool is filled.

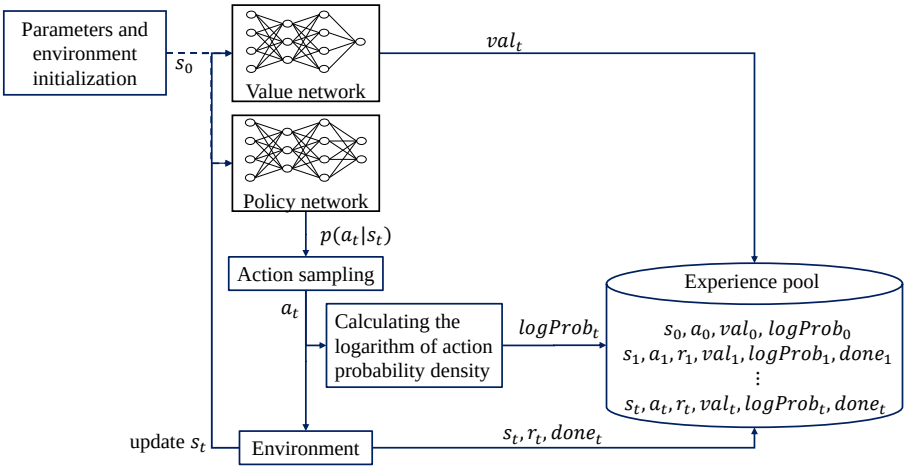

Figure 7. Environment exploration and experience gathering process.

After the experience pool is filled, the data in the experience pool is then used to update the policy and the value network parameters. The updating process is shown in Figure 8 with the following steps:

1. The target output val_{Tar} of the value network is calculated as shown in Formula (2). It means the discounted cumulative reward from the current moment t to the end moment T of the episode, which can be calculated using the rewards of each moment in the experience pool.

$$val_{Tar}(s_t) = \sum_{n=t}^{T} \gamma^{n-1} r_n \tag{2}$$

where γ is the discount factor, which is usually set to 0.95–0.995.

2. The state in the experience pool is input to the value network and the policy network, respectively. The value network outputs the estimate val_{New} of state value, and the policy network outputs the action probability $p_{New}(a|s)$. The specific actions a_{New} are obtained after sampling by action probability. Then the logarithm of action probability density $logProb_{New}$ is calculated.
3. The advantage function $A(s_t)$ is calculated from r_t, $val(s_t)$ and $val(s_{t+1})$ provided by the experience pool, as shown in Formula (3).

$$A(s_t) = \sum_{n=t}^{T} \{\gamma\lambda)^{n-1}(r_t + \gamma \cdot [val(s_{t+1}) - val(s_t)]\} \tag{3}$$

where λ is the attenuation factor, which is usually set to 0.95–0.995.

4. Minimizing the squared difference between all val_{Tar} and val_{New} in a single training batch B is taken as the value network optimization objective J_{cri} (as shown in Formula (4)). The value network parameters are optimized by the Adam optimization method, and then the parameters of the value network are updated. Substitute $logProb_{New}$, $logProb_t$, and $A(s_t)$ in the batch into the policy network optimization objective as shown in Formulas (5) and (6), and optimize the policy network parameters by Adam optimization method to update the parameters of the policy network.

$$J_{cri} = \sum_{s_t \in B} [val_{Tar}(s_t) - val_{New}(s_t)]^2 \tag{4}$$

$$J_{act} = \sum_{s_t \in B} \min\left\{ \frac{p_{\theta'}(s_t|a_t)}{p_\theta(s_t|a_t)} \cdot A(s_t), clip\left(\frac{p_{\theta'}(s_t|a_t)}{p_\theta(s_t|a_t)}, \epsilon \right) \cdot A(s_t) \right\} \tag{5}$$

where $\frac{p_{\theta'}(s_t|a_t)}{p_\theta(s_t|a_t)}$ is importance weight, representing the degree of difference between policy $p_{\theta'}$ and policy p_θ. During each update of the policy, the importance weight is calculated as shown in Formula (6). The $clip$ is a saturation function, as shown in Formula (7), that can restrict a variable x to be between $1-\epsilon$ and $1+\epsilon$. ϵ is usually taken as 0.05–0.2.

$$\frac{p_{\theta'}(s_t|a_t)}{p_\theta(s_t|a_t)} = \frac{logProb_{New}}{logProb_t} \tag{6}$$

$$clip(x, \epsilon) = \begin{cases} 1-\epsilon & x < 1-\epsilon \\ x & 1-\epsilon \le x \le 1+\epsilon \\ 1+\epsilon & x > 1+\epsilon \end{cases} \tag{7}$$

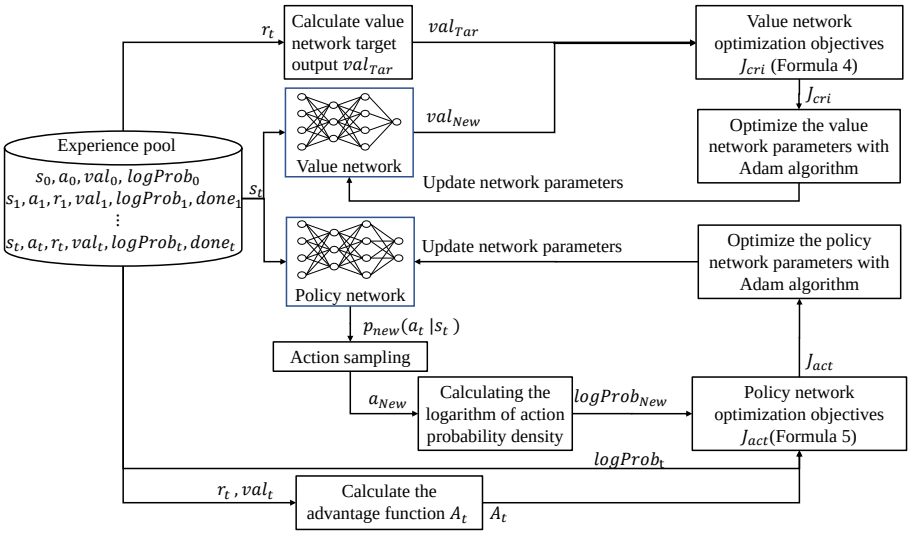

Figure 8. Experience replay and network parameters update process.

After updating the network parameters several times, the experience pool needs to be cleared. The updated networks are used for experience collection to form a new experience pool. Repeatedly and alternatively, experience collection and network parameter updates are performed to gradually approximate the globally optimal policy network parameters.

4.2. Training Environment Construction

The RL environment can help agents gain experience by interacting with policies and providing feedback. We built the RL environment as shown in Figure 9. The output value of the policy network is between −1 and 1, so it needs to be inverse normalized to obtain the real action value. The action variables obtained after the inverse normalization are the acceleration and the yaw rate. Then the vehicle model updates and outputs the vehicle state after receiving acceleration and yaw rate inputs, while the state of surrounding vehicles is updated by the dataset NGSIM in time. The updated vehicle state and the surrounding vehicle state constitute the environment state. Whether the current episode is finished or not is indicated by flag $done_t$, which is determined by the state of the environment. It is considered as finished when one of the following conditions is met: one is that the vehicle states violate the safety constraints (e.g., collision with surrounding vehicles, driving out of the outermost lane, etc.) and the other is that the vehicle travels to the end of the road. Next, the reward r_t is calculated based on the environment states and flag $done_t$. Finally, the flag $done_t$, reward r_t, and the normalized environment states s_t are output.

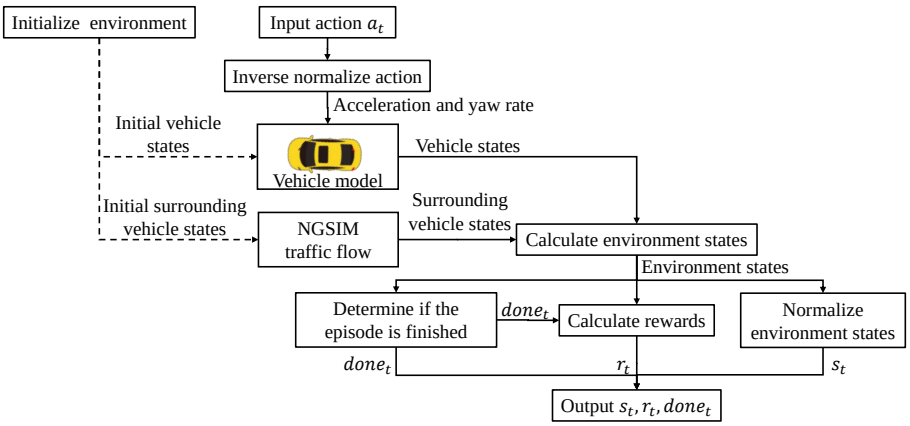

Figure 9. Interactive environment for reinforcement learning.

In the above steps, the vehicle model and the selection of the environment state are crucial. The vehicle model contains two parts: a discrete kinematic differential model and an energy consumption model. The kinematic differential model is shown in Formula (8).

$$\begin{cases} yaw_{veh}(k+1) = yaw_{veh}(k) + yawRate_{veh}(k) \cdot \Delta t \\ v_{veh}(k+1) = v_{veh}(k) + a_{veh}(k) \cdot \Delta t \\ vx_{veh}(k+1) = v_{veh}(k) \cdot \sin(yaw_{veh}(k)) \\ vy_{veh}(k+1) = v_{veh}(k) \cdot \cos(yaw_{veh}(k)) \\ x_{veh}(k+1) = x_{veh}(k) + vx_{veh}(k) \cdot \Delta t \\ y_{veh}(k+1) = y_{veh}(k) + vy_{veh}(k) \cdot \Delta t \end{cases} \tag{8}$$

where $a_{veh}(k)$ and $yawRate_{veh}(k)$ are the acceleration and yaw rate of the input at step k, respectively; yaw_{veh} and v_{veh} are the yaw angle and velocity of the vehicle, respectively; vx_{veh} and vy_{veh} are the components of the vehicle velocity in the lateral and longitudinal directions; x_{veh} and y_{veh} are the vehicle positions, and Δt is the time step of each interaction, which is 0.1 s in this paper.

The resistance to be overcome when the vehicle is moving mainly includes rolling resistance F_f, grade resistance F_i, acceleration resistance F_j, and air resistance F_w, as shown in Formula (9).

$$\begin{cases} F_f = m \cdot g \cdot f \cdot \cos\alpha \\ F_i = m \cdot g \cdot \sin\alpha \\ F_j = \delta \cdot m \cdot a_{veh} \\ F_w = \dfrac{C_D \cdot A_f \cdot \rho \cdot v_{veh}^2}{2} \end{cases} \tag{9}$$

where m is the mass of the vehicle, g is the gravitational acceleration, f is the rolling resistance coefficient, α is the slope of the road, δ is the rotating mass conversion factor, C_D is the air resistance coefficient, A_f is the windward area, and ρ is the air density. The required driving force of the vehicle can be calculated by Formula (10).

$$F_d = F_f + F_i + F_j + F_w \tag{10}$$

The drive force is transmitted to the tires by the power motor through the drivetrain, so the required output torque T_m of the power motor can be calculated by Formula (11).

$$T_m = \frac{F_d r_w}{i_g \eta_T} \tag{11}$$

where i_g represents the ratio of the main gearbox, η_T represents the efficiency of the driveline, and r_w is the tire radius. The angular speed ω_m and revolutions per minute (RPM) n_m at which the power motor runs can be calculated from the vehicle speed v_{veh}, as shown in Formula (12).

$$\begin{cases} \omega_m = \dfrac{v_{veh} i_g}{r_w} \\ n_m = \dfrac{60\omega_m}{2\pi} \end{cases} \tag{12}$$

As shown in Formula (13), the demanded power P_m of the power motor can be obtained from the angular speed ω_m and torque T_m.

$$P_m = \begin{cases} \dfrac{\omega_m T_m}{\eta_m} & T_m \geq 0 \\ \omega_m T_m \eta_m & T_m < 0 \end{cases} \tag{13}$$

where $T_m \geq 0$ means that the motor is in drive mode, $T_m < 0$ means that the motor is in energy recovery mode, eta_m is the efficiency of the motor, and eta_m is estimated by using a map. According to the experimental data, the motor efficiency map shown in Figure 10 is obtained.

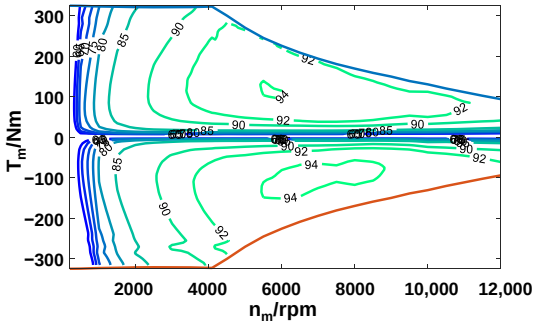

Figure 10. Motor efficiency map.

By looking up the map shown in Figure 10, the efficiency η_m of the motor at the operating point can be obtained through T_m and n_m. Then the demand power of the motor can be calculated by substituting η_m into Formula (13). The energy consumed $E_m(k)$ can be calculated by integrating the motor power P_m over time, as shown in Formula (14).

$$E_m(k) = E_m(k-1) + P_m(k)\Delta t = \sum_{i=0}^{k} P_m(i)\Delta t \tag{14}$$

The vehicle model parameters used are shown in Table 3.

Table 3. Vehicle model parameters.

Variables	Value	Variables	Value
m/kg	1845	$\rho/(kg \cdot m^{-3})$	1.2
r_w/m	0.335	i_g	9.7
C_D	0.29	α	0
f	0.01	δ	1.1
$g/(m \cdot s^{-2})$	9.8	η_T	0.98
A_f/m^2	2.48	-	-

To verify the accuracy of the established energy consumption model, the model was compared with the software CRUISE, and the simulation results are shown in Figure 11. The figure shows that the simulation results of the model we built are close to those of the CRUISE software model. In addition, the CRUISE software shows that the average electric consumption of our car under NEDC conditions is 14.37 kWh/100 km, while the result obtained from the model we built is 13.54 kWh/100 km, with an overall error within 6%, which proves that the energy consumption model we built is accurate.

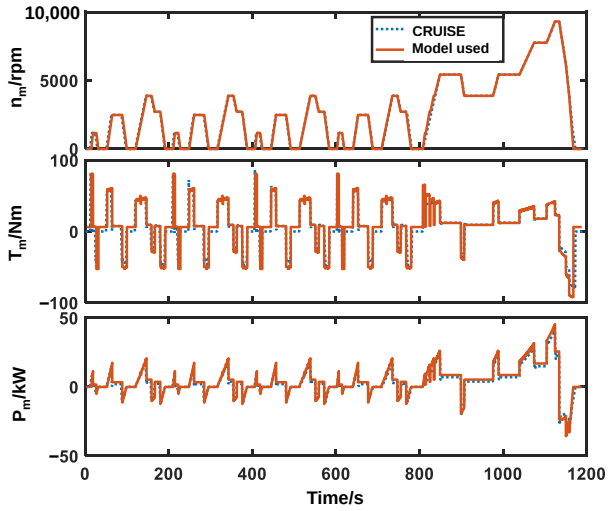

Figure 11. Energy consumption model simulation comparison.

The state vector consists of multiple states. In terms of the ego vehicle states, the state vector contains the velocity, acceleration, yaw, and lane number of the ego vehicle. In terms of surrounding vehicles states, the state vector contains the states of the nearest vehicles in front of and behind the ego vehicle in the lane where the ego vehicle is located and in the two adjacent lanes, for a total of six vehicles. The states information of each surrounding vehicle includes the longitudinal and lateral relative distance between the vehicle and ego vehicle, the longitudinal and lateral velocity, the longitudinal acceleration at the current moment, and the longitudinal and lateral relative distance between the vehicle and ego vehicle in the next 5 s, for a total of 15 states. In summary, the environment is described by 4 states of the ego vehicle and 60 states of the six surrounding vehicles, for a total of 94 states.

4.3. Reward Function Design and Model Training

The reward function of energy-efficient driving must consider the constraints of safety, comfort, and practicality. The design of the reward function is divided into immediate rewards and settlement rewards. Immediate rewards are obtained at each interaction step and serve to evaluate each step, thus inspiring and guiding the policy to optimize in the desired direction, while settlement rewards are given at the end of the episode and serve to evaluate the whole episode, thus guiding the policy to consider global optimality.

For the safety constraint, firstly, the ego vehicle cannot drive out of the outermost lane, otherwise the episode is judged to be over and the penalty value r_{f1} for failure is added to the reward function r_t; secondly, the ego vehicle cannot collide with the surrounding vehicles, otherwise the episode is also judged to be over and the penalty value r_{f2} for failure is added to the reward function r_t.

For all finished episodes, a progress reward r_p is given proportional to the progress completed at the end of the episode, and the closer to the end of the road the higher the reward. Each interaction that does not fail is rewarded with a safety reward r_s.

In order to satisfy the comfort constraints, it is necessary to restrict the boundaries of the action inputs, i.e., the inputs should satisfy $|a_{veh}| < A_{max}$, $|yaw_{veh}| < yaw_{max}$.

To increase the travel efficiency, the time consumption penalty r_{tc} is added to the instantaneous reward function r_t, and the more time consumed, the larger the accumulated penalty value. At the same time, a speed reward r_v proportional to the instantaneous speed is given, inspiring the policy to try to output the action that makes the velocity increase.

The energy-saving objective is considered in the settlement reward. When the whole episode is finished, the energy consumption per 100 km is calculated for the whole episode and a penalty value r_{ec} is given, the higher the energy consumption, the higher the penalty value.

In summary, the entire reward function can be expressed by Formula (15). When $done_t = 0$, it means the current episode is not finished, and returns the instantaneous reward, which is the sum of the safety reward r_s, time consumption penalty r_{tc}, and speed reward r_v. When $done_t = 1$, it means the current episode is finished, and the settlement reward is added to the instantaneous reward, which returns the sum of time consumption penalty r_{tc}, speed reward r_v, progress reward r_p, energy consumption penalty r_ec, and safety penalty r_{f1}, r_{f2} or reward r_s.

$$r_t = \begin{cases} r_s + r_{tc} + r_v & done_t = 0 \\ r_{tc} + r_v + r_p + r_{ec} + f_1 r_{f1} + f_2 r_{f2} + (1 - f_1)(1 - f_2) r_s & done_t = 1 \end{cases} \quad (15)$$

where f_1 and f_2 are Boolean values that represent whether the ego vehicle in the current episode has driven out of the outermost lane or collided with other vehicles. The rewards values used for the network training in this paper are shown in the Table 4.

Table 4. The rewards values used for the network training.

r_s	r_{tc}	r_v	r_p	r_{ec}	r_{f1}	r_{f2}
0.01	−0.02	0.01	35	−0.5	−40	−40

As shown in Figure 12, based on the input dimensions of the vehicle model and the output dimensions of the state feedback described before, the input layer dimension of the policy network is determined to be 94 and the output layer dimension is 2. The input layer dimension of the value network is 94 and the output layer dimension is 1. Three dense layers are selected as hidden layers. The activation function of the first two layers is the Relu function, and that of the last layer is the Hardswitch function.

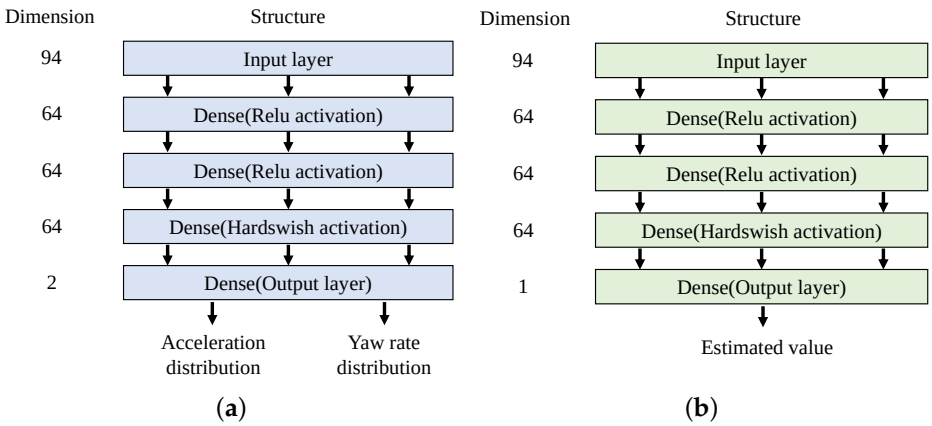

Figure 12. Artificial neural network structure: (**a**) policy network structure. (**b**) Value network structure.

The policy network is trained according to the above network structure and parameter settings, and the convergence curve is shown in Figure 13. During the training process, the parameters are updated 2^8 times after collecting 2^{14} frames of experience data, and the average reward under the current network parameters, as well as the historical maximum reward are recorded. The model parameters that produce the historical maximum reward are saved and used as the parameters of the final output policy network model.

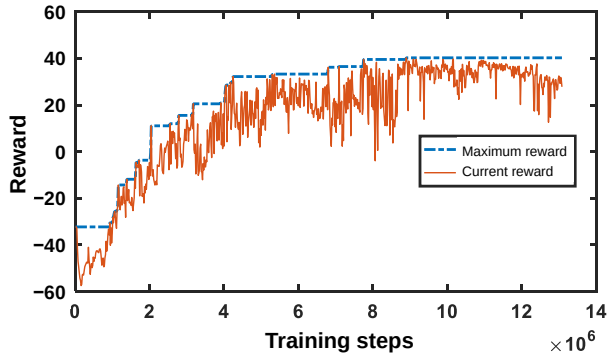

Figure 13. Reward function convergence curve.

5. Results and Discussions

5.1. Simulation Results

To verify the energy-saving effect of the proposed method, the method is simulated and compared with other methods. Three main methods are included, firstly a rule-based planning method, secondly a RL method that does not consider energy saving, and, finally, an RL method that considers energy saving but does not use prediction information. A frame from the NGSIM dataset is randomly extracted as the starting frame for the simulation, and the ego vehicle is placed into the environment for testing. After reaching the end of the road, the vehicle decelerates to zero with a deceleration speed of 0.2 m/s^2 for kinetic energy recovery. When the vehicle speed decelerates to 0, a frame is randomly selected as the starting frame, and the vehicle position is reset as the starting point to start a new test. The test was repeated 10 times, forming a test distance of about 5 km. The comparison results of the four methods are shown in Figures 14–16.

Figure 14 shows that the rule-based approach yields a larger variation in acceleration, as well as several large negative values compared to the RL approach. The acceleration of the trajectories obtained by the rule-based method is less than -2 m/s^2 around 20 s, 260 s, 350 s, 480 s, 600 s, 630 s, and 700 s. It means that the vehicle has frequent braking, which is not only less comfortable, but also increases unnecessary energy consumption. Compared to other RL-based methods, our method has a significantly lower acceleration variation frequency and, therefore, better comfort.

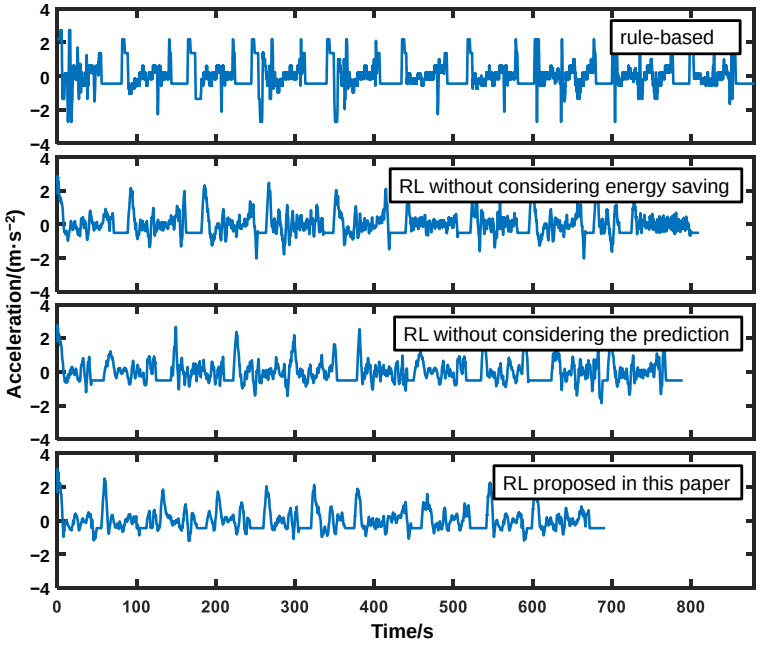

Figure 14. Acceleration profile comparison.

Figure 15 shows that the rule-based method has a more fixed torque distribution, which is due to the fact that the method uses several different fixed accelerations switched by some logical thresholds. The RL model that does not consider energy conservation has more operating points located in the region with higher torque, indicating that the model policy obtains more trajectories with sharp acceleration and deceleration. The operating points of our model are less distributed in the high torque region, indicating that the accelerations of trajectories are smaller, which is beneficial to energy saving.

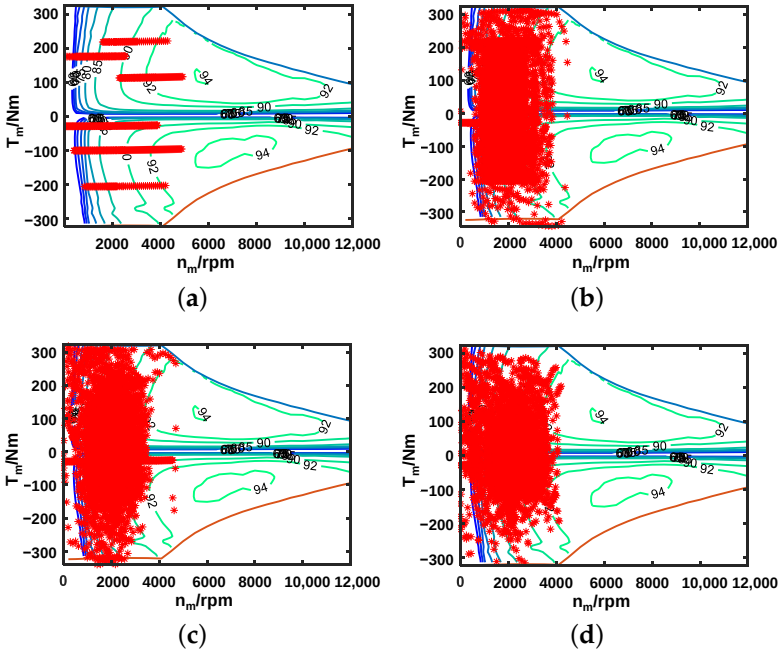

Figure 15. Distribution of motor operating points: (**a**) rule-based. (**b**) RL without considering energy saving. (**c**) RL without considering the prediction. (**d**) RL proposed in this paper.

Figure 16 shows that the model with the highest energy consumption per kilometer is the RL model that does not consider energy saving, followed by the rule-based approach, and then the RL model that does not use prediction information. It can be seen that the energy-consumption penalty term and the prediction information introduced into the RL model in this paper are helpful to improve the energy efficiency.

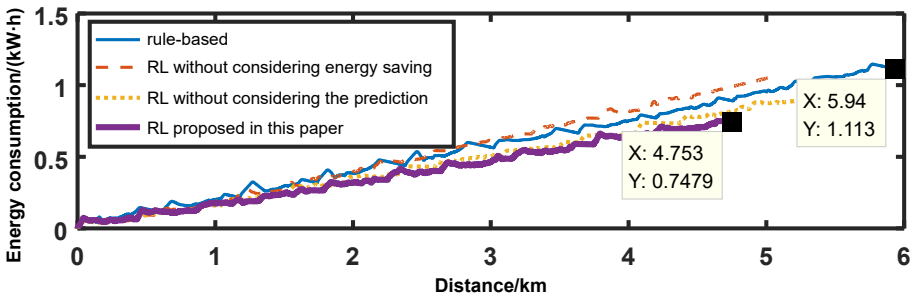

Figure 16. The comparison of vehicle energy consumption for different methods.

Table 5 shows the comparison of key indicators of self-vehicle trajectory using different methods. As can be seen from the average speeds in the table, the RL model that does not consider energy saving has a slightly higher average speed. From the acceleration variance, it can be seen that the rule-based approach has a more drastic variation in acceleration, which may be due to the lack of smooth transitions between rules. The acceleration variance of the RL-based approach is relatively small. The proposed model has the lowest acceleration variance, which implies better comfort.

The comparison of energy consumption per 100 km shows that our model has the best energy efficiency, reducing energy consumption by 9.3% compared to the model without prediction information, 21.9% compared to the model without considering energy efficiency, and 16.1% compared to the rule-based model. The comparison of the indicators in the table shows that the model proposed is able to save energy consumption without affecting the travel efficiency and comfort.

Table 5. Comparison of trajectory characteristics of different methods.

Method	Average Velocity/km· h^{-1}	Acceleration Variance/($m^2 \cdot s^{-4}$)	Average Energy Consumption Per 100 km/(kW · h)
Rule-based	24.2	0.6633	18.74
RL without considering energy saving	25.1	0.4382	20.13
RL without considering the prediction	24.3	0.4058	17.34
RL proposed in this paper	24.4	0.3751	15.73

5.2. Discussions

In this paper, we propose an energy-saving driving method. A hierarchical LSTM model is established to predict the trajectory of surrounding vehicles, and then an RL-based energy-efficient driving model is built and trained. Simulation results show that our method has better energy efficiency than rule-based methods, conventional RL methods, and RL energy-saving methods that do not take into account prediction information. There are two main reasons to explain this superiority.

Firstly, a more accurate prediction model is used, resulting in more forward-looking driving behavior. Previous research on energy-efficient driving tends to take a simpler trajectory prediction model for the convenience of solving [23,24,43], which has a large error in predicting vehicle trajectories for the medium and long term. The results of this paper show that a more accurate trajectory prediction model can improve the energy saving potential of driving methods. Surrounding vehicle trajectory prediction can help vehicles avoid speed fluctuations and reduce extra energy consumption due to unnecessary braking, thus improving energy efficiency, which is consistent with the findings of literature [14].

Secondly, the method is able to perform lane changes with the goal of saving energy. Most of the previous studies only considered eco-cruise control on a single lane [32,43,44]. However, the traffic flow conditions in different lanes have a large impact on the speed and acceleration of the vehicle, which affects the energy efficiency. Refs. [34,35] also focused on this issue, thus including the lane as a variable and saving some energy. Further to [34,35], our approach does not ignore the details of the lane change process, but achieves the lane change step by step through the vehicle lateral control variables, thus ensuring the safety of the vehicle lateral motion.

The limitation of our approach is that the scenario is closed, and future work will target more open and pedestrian-involved traffic environments. In addition, as a data-driven method, a large amount of data are required in the model training phase, and the data in some scenarios are difficult to obtain, limiting the wide application of the method.

6. Conclusions

CAVs have greater energy-saving potential due to their advantages in environmental sensing and vehicle control. We propose an energy-efficient driving method from the perspective of multi-lane lateral and longitudinal integrated control.

To improve the forward-looking capability of the algorithm, we construct a medium and long time horizon vehicle trajectories prediction method based on hierarchical LSTM neural networks. The trajectory prediction task is decomposed into three steps: driving intention decision prediction, lane change time prediction, and trajectory prediction. Mul-

tiple LSTM networks are built and trained for different tasks in different steps to form a complete model combination, and the final trajectory prediction is accomplished through reasonable invocation of the three step models. The test results on the NGSIM dataset show that the proposed prediction method has significantly lower prediction errors in the medium and long time horizons relative to the constant acceleration model, the polynomial model, and the integral LSTM network.

Combining the trajectory prediction information, an energy-efficient driving method based on PPO reinforcement learning is developed. The energy consumption is considered in the RL reward term, and constraints are imposed on safety, comfort, and traffic efficiency to obtain an energy-efficient driving policy that can take into account multiple constraints. Simulation results show that the method can effectively reduce energy consumption by 9.3–21.9% while taking into account multiple constraints.

Author Contributions: Funding acquisition, H.M. and F.Y.; Coding, X.X. and Y.Y.; Validation, X.X. and P.W.; Writing original draft, X.X. and W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China Youth Fund Project (52102458) and Jilin Province Science and Technology Development Program (20210301023GX).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Post, J.; Veldstra, J.; Ünal, A. Acceptability and Acceptance of Connected Automated Vehicles: A Literature Review and Focus Groups. In Proceedings of the 5th International Conference on Computer-Human Interaction Research and Applications—SUaaVE, Online, 28–29 October 2021; INSTICC, SciTePress: Bach, Switzerland, 2021; pp. 223–231. [CrossRef]
2. Vahidi, A.; Sciarretta, A. Energy saving potentials of connected and automated vehicles. *Transp. Res. Part Emerg. Technol.* **2018**, *95*, 822–843. [CrossRef]
3. Al-Wreikat, Y.; Serrano, C.; Sodré, J. Driving behaviour and trip condition effects on the energy consumption of an electric vehicle under real-world driving. *Appl. Energy* **2021**, *297*, 117096. [CrossRef]
4. Zheng, F.; Li, J.; Zuylen, H.V.; Lu, C. Influence of driver characteristics on emissions and fuel consumption. *Transp. Res. Procedia* **2017**, *27*, 624–631. [CrossRef]
5. Jeffrey, G.; Matthew, E.; Witt, S. Analyzing Vehicle Fuel Saving Opportunities through Intelligent Driver Feedback. *SAE Int. J. Passeng. Cars—Electron. Electr. Syst.* **2012**, *5*, 450–461.
6. Hari, D.; Brace, C.J.; Vagg, C.; Poxon, J.; Ash, L. Analysis of a Driver Behaviour Improvement Tool to Reduce Fuel Consumption. In Proceedings of the International Conference on Connected Vehicles & Expo, Las Vegas, NV, USA, 2–6 December 2013.
7. Vagg, C.; Brace, C.J.; Hari, D.; Akehurst, S. Development and Field Trial of a Driver Assistance System to Encourage Eco-Driving in Light Commercial Vehicle Fleets. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 796–805. [CrossRef]
8. Magana, V.C.; Munoz-Organero, M. GAFU: Using a gamification tool to save fuel. *IEEE Intell. Transp. Syst. Mag.* **2015**, *7*, 58–70. [CrossRef]
9. Jinjian, L.I.; Dridi, M.; El-Moudni, A. Multi-vehicles green light optimal speed advisory based on the augmented lagrangian genetic algorithm. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Qingdao, China, 8–11 October 2014.
10. Lee, J.; Nelson, D.J.; Lohse-Busch, H. Vehicle Inertia Impact on Fuel Consumption of Conventional and Hybrid Electric Vehicles Using Acceleration and Coast Driving Strategy. In Proceedings of the SAE World Congress & Exhibition, Detroit, MI, USA, 20–23 April 2009.
11. Tian, Z.; Liu, L.; Shi, W. A pulse-and-glide-driven adaptive cruise control system for electric vehicle. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e13054. [CrossRef]
12. Shieh, S.Y.; Ersal, T.; Peng, H. Pulse-and-Glide Operation for Parallel Hybrid Electric Vehicles with Step-Gear Transmission in Automated Car-Following Scenario with Ride Comfort Consideration. In Proceedings of the 2019 American Control Conference, Philadelphia, PA, USA, 10–12 July 2019; pp. 959–964.
13. Xu, N.; Li, X.; Liu, Q.; Zhao, D. An Overview of Eco-Driving Theory, Capability Evaluation, and Training Applications. *Sensors* **2021**, *21*, 6547. [CrossRef]
14. Manzie, C.; Watson, H.; Halgamuge, S. Fuel economy improvements for urban driving: Hybrid vs. intelligent vehicles. *Transp. Res. Part Emerg. Technol.* **2007**, *15*, 1–16. [CrossRef]
15. Wan, N.; Zhang, C.; Vahidi, A. Probabilistic Anticipation and Control in Autonomous Car Following. *IEEE Trans. Control. Syst. Technol.* **2019**, *27*, 30–38. [CrossRef]

16. Mahler, G.; Vahidi, A. An Optimal Velocity-Planning Scheme for Vehicle Energy Efficiency Through Probabilistic Prediction of Traffic-Signal Timing. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2516–2523. [CrossRef]
17. Sun, C.; Hu, X.; Moura, S.J.; Sun, F. Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles. *IEEE Trans. Control. Syst. Technol.* **2015**, *23*, 1197–1204. [CrossRef]
18. Xu, S.; Zhang, C.; Wang, H.; Xu, S.; Liu, W. Research on energy-saving approach of multi-modal vehicles based on traffic signal control. In Proceedings of the 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 26–28 November 2017; pp. 1–6. [CrossRef]
19. Wang, X.; Park, S.; Han, K. Energy-Efficient Speed Planner for Connected and Automated Electric Vehicles on Sloped Roads. *IEEE Access* **2022**, *10*, 34654–34664. [CrossRef]
20. Weißmann, A.; Görges, D.; Lin, X. Energy-optimal adaptive cruise control combining model predictive control and dynamic programming. *Control. Eng. Pract.* **2018**, *72*, 125–137. [CrossRef]
21. Kamal, M.A.S.; Imura, J.i.; Hayakawa, T.; Ohata, A.; Aihara, K. Smart Driving of a Vehicle Using Model Predictive Control for Improving Traffic Flow. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 878–888. [CrossRef]
22. Pan, C.; Huang, A.; Wang, J.; Chen, L.; Liang, J.; Zhou, W.; Wang, L.; Yang, J. Energy-optimal adaptive cruise control strategy for electric vehicles based on model predictive control. *Energy* **2022**, *241*, 122793. [CrossRef]
23. McDonough, K.; Kolmanovsky, I.; Filev, D.; Yanakiev, D.; Szwabowski, S.; Michelini, J. Stochastic dynamic programming control policies for fuel efficient vehicle following. In Proceedings of the 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013; pp. 1350–1355. [CrossRef]
24. Wang, S.; Lin, X. Eco-driving control of connected and automated hybrid vehicles in mixed driving scenarios. *Appl. Energy* **2020**, *271*, 115233. [CrossRef]
25. Deo, N.; Trivedi, M.M. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 1179–1184. [CrossRef]
26. Choi, D.; Yim, J.; Baek, M.; Lee, S. Machine Learning-Based Vehicle Trajectory Prediction Using V2V Communications and On-Board Sensors. *Electronics* **2021**, *10*, 420. [CrossRef]
27. Deo, N.; Trivedi, M.M. Convolutional Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1549–15498. [CrossRef]
28. Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Non-local Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 975–980. [CrossRef]
29. Dai, S.; Li, L.; Li, Z. Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction. *IEEE Access* **2019**, *7*, 38287–38296. [CrossRef]
30. Chen, G.; Hu, L.; Zhang, Q.; Ren, Z.; Gao, X.; Cheng, J. ST-LSTM: Spatio-Temporal Graph Based Long Short-Term Memory Network For Vehicle Trajectory Prediction. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Online, 25–28 October 2020; pp. 608–612. [CrossRef]
31. Nie, Z.; Farzaneh, H. Adaptive Cruise Control for Eco-Driving Based on Model Predictive Control Algorithm. *Appl. Sci.* **2020**, *10*, 5271. [CrossRef]
32. Bakibillah, A.S.M.; Kamal, M.A.S.; Tan, C.P.; Hayakawa, T.; Imura, J.-i. Fuzzy-tuned model predictive control for dynamic eco-driving on hilly roads. *Appl. Soft Comput.* **2021**, *99*, 106875. [CrossRef]
33. Kamal, M.A.S.; Taguchi, S.; Yoshimura, T. Efficient vehicle driving on multi-lane roads using model predictive control under a connected vehicle environment. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; pp. 736–741. [CrossRef]
34. Kamal, M.A.S.; Taguchi, S.; Yoshimura, T. Efficient Driving on Multilane Roads Under a Connected Vehicle Environment. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2541–2551. [CrossRef]
35. Tajeddin, S.; Ekhtiari, S.; Faieghi, M.; Azad, N.L. Ecological Adaptive Cruise Control With Optimal Lane Selection in Connected Vehicle Environments. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4538–4549. [CrossRef]
36. FHWA. The Next Generation Simulation Program. Available online: http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm (accessed on 13 June 2022).
37. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef]
38. Wu, X.; Gao, X.; Zhang, W.; Luo, R.; Wang, J. Learning over Categorical Data Using Counting Features: With an Application on Click-through Rate Estimation. In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data, Anchorage, AK, USA, 5 August 2019; DLP-KDD'19; Association for Computing Machinery: New York, NY, USA, 2019. [CrossRef]
39. Kline, D.M.; Berardi, V.L. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Comput. Appl.* **2005**, *14*, 310–318. [CrossRef]

40. Agostinelli, F.; Hocquet, G.; Singh, S.; Baldi, P. From Reinforcement Learning to Deep Reinforcement Learning: An Overview. In Proceedings of the Braverman Readings in Machine Learning. Key Ideas from Inception to Current State: International Conference Commemorating the 40th Anniversary of Emmanuil Braverman's Decease, Boston, MA, USA, 28–30 April 2017, Invited Talks; Rozonoer, L., Mirkin, B., Muchnik, I., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 298–328. [CrossRef]
41. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 1928–1937.
42. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**.
43. Han, J.; Sciarretta, A.; Ojeda, L.L.; De Nunzio, G.; Thibault, L. Safe- and Eco-Driving Control for Connected and Automated Electric Vehicles Using Analytical State-Constrained Optimal Solution. *IEEE Trans. Intell. Veh.* **2018**, *3*, 163–172. [CrossRef]
44. Li, J.; Liu, Y.; Zhang, Y.; Lei, Z.; Chen, Z.; Li, G. Data-driven based eco-driving control for plug-in hybrid electric vehicles. *J. Power Sources* **2021**, *498*, 229916. [CrossRef]