Received: 4 November 2021 | Revised: 1 June 2022 | Accepted: 22 September 2022

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12286

The Institution of Engineering and Technology

WILEY

ORIGINAL RESEARCH

Improving traffic signal control operations using proximal policy optimization

Liben Huang | Xiaohui Qu

School of Electrical Engineering, Southeast University, Nanjing, China

Correspondence
Xiaohui Qu, School of Electrical Engineering, Southeast University, Nanjing 210096, China.
Email: xhqu@seu.edu.cn

Funding information
Industrial Chain Demonstration Application Scenario Construction Project of Nanjing Industrial and Informatization Development Special Funds in 2020, Grant/Award Number: 012947218/2020-133671

Abstract

Existing traffic signal control systems present many limitations including fixed signal timing schemes, insufficient efficiency and flexibility, and difficulty in adapting to the changing traffic flows. In recent years, the development of deep reinforcement learning (RL) has shown great research potential and application prospects. This paper proposes an intersection signal control method based on the proximal policy optimization (PPO) method. Specifically, this paper uses the value vector representation method of traffic characteristics to encode the traffic state and then feeds the state encoding into the long short-term memory (LSTM) network to obtain the signal phase output. Finally, to obtain the optimal signal control strategy, the PPO algorithm is utilized to train the neural network and adjust the signal phase. The proposed algorithm is benchmarked against other classic RL and adaptive signal control schemes in an experimental environment based on the traffic simulation software simulation of urban mobility (SUMO). Experimental results showed that the proposed algorithm greatly improves traffic efficiency. Specifically, the mean velocity of the vehicles increases by 45.09%, and the mean occupancy rate of each lane, the length of the longest jams during each step, and the mean halting duration dropped by 21.38%, 25.86%, and 12.94%, respectively.

1 | INTRODUCTION

Traffic signal control is an important subject in urban traffic management control. Reasonable traffic signal control separates the time dimension of the conflicting traffic flows in space, maximizing thus the space-time operation efficiency of the traffic flow, and playing an important role in alleviating traffic congestion [1] by improving network throughput and operation efficiency, and by reducing vehicle emissions and fuel consumption. Therefore, the analysis of the temporal-spatial traits of intersection traffic flow and the intelligent selection of the signal control strategies to adapt to the real-time changing traffic demands have attracted increasing research interest.

Since the 1960s, different scholars have put forward different kinds of signal control methods, whose development course can be divided into four main stages. The first stage is the controller with a fixed time. In 1958, Webster proposed a fixed-time control method based on road traffic to calculate the optimal signal cycle length and the time of green light at different phases [2]. As one of the earliest traffic signal control methods, fixed-time control can reasonably estimate the traffic demand according to the observational data of the historical traffic. It can thus calculate the phase duration of the signal in advance but tend to perform poorly in processing dynamic real-time traffic [3]. When faced with multiple intersections, the green wave and the maximum bandwidth achieve the intersection cooperative control mode by calculating the signal offset of the adjacent intersections [4, 5]. Varaiya proposed the concept of the intersection pressure, the maximum pressure control method, and the balanced flow pressure between intersections to minimize the pressure in each phase of the intersection [6]. However, these methods are all based on the fixed-time control mode, which faces difficulties to handle abnormal changes in traffic flow, such as traffic accidents and special events, and adjusting signal control strategy in time.

The actuated control will be the direction of the next phase, which was first introduced in the 1970s. In this phase, the detector is used to obtain the state of the traffic flow. When the gap between vehicles in the queue is greater than a pre-defined threshold the current phase can be extended and switched to other phases of the traffic signal adjustment [7]. Some representative actuated control systems are microprocessor optimised vehicle actuation (MOVA) [8, 9], LHVORA, and self optimizing signal (SOS) [10]. Park proposed the concept of dynamic gap-out based on vehicle infrastructure integration (VII), where the control time of the induction signal is adjusted according to the dynamic gap of the vehicle to reduce its delay [11]. Zheng used this method to predict the flow input at the upstream intersection and the steering ratio at the target intersection to dynamically adjust the parameter of the maximum phase duration of the induction-controlled signal to change between traffic conditions [12]. Although the actuated control is real-time, it is still limited to the preset signal phase. When the traffic flow is sparse, the stability of the actuated control is poor. Meanwhile, the actuated control system does not effectively use the accumulated traffic data to analyze the temporal and spatial relationship of the traffic flow at the intersection, due to the lack of an ability to forecast the traffic conditions, adjustment measures in advance.

Recently, adaptive control has been widely used, which adopts the input using pre-defined signal phase, period length, phase offset, and other control parameters. At the same time, according to the predefined methods for evaluating intersection performance and the acquisition of real-time traffic data, such as upstream traffic flow, queue length, and waiting times by sensors like cameras and radar, it can coordinate and control the signal phase by selecting the dynamic parameters using signalling methods between adjacent intersections. Adaptive control can be divided into model-based method and data-driven method according to different signal control strategy adjustment methods. Representative adaptive control systems of model-based method include sydney coordinated adaptive traffic system (SCATS) [13], split cycle offset optimisation technique (SCOOT) [14] and the traffic network study tool (TRANSYT). SCOOT and SCATS adopt real-time online control, and the system adopts hierarchical control structure. According to real-time traffic flow data, the green signal ratio, phase difference, and cycle are controlled with the objective of lowest vehicle queuing and maximum traffic flow. However, model-based methods still have some problems. (1) The predefined signal phase, which can affect strategy iteration on the search domain, is required because the combinatorial space of signal strategy is huge. (2) Extensive field testing and manual adjustments are required to reflect the characteristics of the road and the traffic, and this process requires customized adjustments since it is resource-consuming and not universal. (3) The system relies on parameter setting using prior domain expertise, because of the complexity of the urban traffic system and the lack of a uniform specification for the selection of parameter values. Primarily, different settings may lead to large deviations between the output strategy and the optimal solution.

Multi-source real-time traffic status data can now be collected from urban road traffic networks because of the widespread use of infrastructure data collection equipment. The rich real-time traffic data and the continuous advancements in computer intelligence have facilitated the significant improvement of the effectiveness of traffic signal control. Specifically, reinforcement learning (RL) has been successfully applied in solving complex nonlinear making it an important research direction of intelligent traffic signal control [15, 16]. In 1997, Thorpe applied RL to traffic signal control for the first time [17], and people began to realize that RL provided a new way of thinking for complex non-linear and uncertain networks. Mannion et al. showcase the developing multi agent traffic control architecture in this field. Through Single Agent Reinforcement Learning Approaches, Multi Agent Reinforcement Learning Approaches, and Coordinated multiagent reinforcement learning (MARL) Approaches, the feasibility and challenges of RL and game theoric approaches in traffic control are expounded [18]. Abdoos et al. think that conventional traffic signal timing methods do not result in an efficient control, they model a relatively large traffic network as a multi-agent system and use techniques from multi-agent RL, a parametric representation of the action space has made the method extendable to different types of intersection. The simulation results demonstrate that the proposed Q-learning outperformed the fixed time method under different traffic demands [19]. Instead of designing and extracting the high-dimensional features from the front-end detector data, Shabestary directly transferred the sensor data into a deep neural network and then optimized the signal processing strategy through the continuous interaction between the RL agent and a single intersection [16]. Liang et al. predicted traffic state changes from imaging data, creating a grid of the actual intersection and representing using sparse matrix encoding the traffic state using the relative position of vehicles [20]. Some scholars have proposed a method to estimate the optimal action value function by using recursive neural network (RNN) to process continuous observation series data. Choe et al. found that the Deep Q Network (DQN) model based on RNN was superior to the RL model based on the convolutional neural network (CNN) structure in the signal control of a single intersection, because the analysis and extraction of traffic data time series features were conducive to the selection of traffic signal strategies [21]. However, when the existing value-based methods are faced with the problem of continuous selection of signal control phases with long time series, it is difficult to accurately estimate the value of signal strategy because the intersection traffic state space and the signal phase combination space show exponential explosive growth over time, and the reward distribution is sparse. Although other algorithms including DQN use convolutional neural network and other methods to deal with the problem of large dimension of state space, traffic signal control has delayed effect on traffic flow, and the calculation of action value by the time-difference method will produce over estimation and unstable convergence. Compared with the value-based RL method, the policy-based method is more suitable for solving problems with high-dimensional action space or continuous action space. The policy-based method is to parameterize the policy and update the parameters through the gradient iteration of the action policy. Because the policy can be directly iterated, it has better convergence. The proximal policy optimization (PPO) algorithm which is based on Monte Carlo search, adopts off-line learning and importance sampling to improve the speed of policy convergence while maintaining the learning of random policies, which greatly increases the robustness of policies. Ma et al. defined vehicle states using discrete traffic state coding (DTSE) method, and dynamically adjusted signal strategies using PPO algorithm [35]. However, the policy including the PPO-based method for the representation of the feature space is not complete, traffic status, and convolution neural network was adopted as a strategy network although it can capture the vehicle position information, but ignores the non-linear characteristics of traffic flow in time and space perception, difficult to form the state of traffic flow from the space-time dimension to the signal phase strategy of effective mapping. Therefore, it is easy to fall into the local optimal solution of signal strategy, fixed strategy distribution, poor convergence stability, and other problems.

In order to improve the efficiency of intersections, solve the problems of incomplete expression of traffic flow characteristics at intersections and low perception ability of traffic flow space-time characteristics. In this paper, we introduce a single-point intersection signal control model (LSTM-PPO) integrating PPO [22] and Long Short-Term Memory (LSTM) [23] network. The near-end strategy optimization mainly consists of two steps: Actor and Critic. The Actor step outputs the selection probability of different signal phases according to the traffic state, while the Critic one evaluates the expected value of the action in the traffic state based on the input of the traffic state. Both steps use the LSTM structure to process the input time-series data, and effectively extract the time-series features from the traffic data. The time-series features are effectively extracted from the traffic data with this method. Self-learning is a signalling strategy, which can maximize the efficiency of the intersection and realize intelligent adaptive signal control in the process of constant interaction between the LSTM-PPO model and the intersection environment.

2 | BACKGROUND AND RELATED WORK

With the continuous improvement of simulation model accuracy and computing power, simulation software has been enabled to reproduce real-world traffic behaviour, such as vehicle following and lane change, the formation of traffic jams, and others. Moreover, the simulation of the altitude of a real traffic environment is enabled by the RL algorithm, which adjusts the output of the action strategy through environmental feedback allowing it to be applied to the traffic signal control problem. Therefore, many researchers have been developing traffic signal control algorithms based on RL for decades, and this category of methods has gradually attracted increasing academic attention.

When applying the RL algorithm to traffic signal control, for modelling queue length, vehicle position, vehicle speed, and other parameters, it is essential to encode the traffic status index data and input it into the agent. Then, the agent outputs the signal that should be taken to control the phase according to the policy function. After the signal control phase is implemented, the agent finally collects the change of traffic state and calculates the evaluation value corresponding to the new traffic state. The latter is in turn the reward obtained by adopting the signal control phase. The agent can find an optimal strategy based on changes in traffic conditions and feedback rewards to minimize congestion and maximize traffic efficiency at intersections through all the circulation steps. Therefore, according to the different control steps of the algorithm, the researchers have mainly studied four areas: state space, action space, reward, and model building.

First of all, the RL algorithms work only if an accurate and clear definition of the state space exists. Therefore, many scholars have adopted the representation of different states in their studies. Mousavi, Genders, and others, being inspired by the representation method of RGB image, obtained the state matrix, have captured the real-time images from software simulation, and encoded the position of the vehicle using the binary method [24, 25]. This method can convert continuously observed traffic flow data into discrete images and is therefore called DTSE. The DTSE divides the simulated intersection according to lanes, with each lane being cut into N cells with the standard vehicle length, and one or more features such as vehicle position, speed, and acceleration being filled into the cells, formulating thus a state matrix [26, 27].

Vector way based on feature is another method for state representation [28–30]. Lt is different from the method of discrete statistics of vehicle data. In this method, the running data of independent vehicles is statistically analyzed, including the calculation of queue length, the average waiting time, the lane average speed, the traffic flow, and other characteristic data. Then the feature vector of state representation is constructed using the statistical data of each intersection lane.

After the agent receives the status input, it requires the selection of an action to execute from the set of all available actions. Thus, the behaviour of the output is acting as the bridge between the agent and the environment. For a four-way intersection, each direction is composed of three phases: green, yellow, and red. The action mode definition can be roughly divided into three types. The first is the choice of the phase direction from the green light. When a fixed moment is experienced every time, the agent chooses in one direction to perform the green light phase, and in all the other directions to perform the red-light phases. Early studies, such as Van Der Pol [31], defined the actions of agents only as of the green light phase in the east-west direction (EW) and the green light phase in the north-south direction (NS). With the improvement of traffic control requirements at intersections, the mode of phase combination has become more diverse, including an action set of the four phases of north-south straight (NSS), north-south left-turn green (NSL), east-west straight green (EWS), and east-west left-turn (EWL) as a relatively common combination [32].

The third motion mode, in contrast to the previous two discrete motion selection methods, is based on the continuous motion space method, which acts under the condition of fixed phase sequence, predicting the duration of the next phase. At the same time, Casas [29] preset the maximum and minimum values of phase duration to ensure that the predicted value will not deviate from the actual control requirements.

The rewards given by intelligent decency in different states to directly determine the value of action choices are the decisive factor that distinguishes RL algorithms from the other types of machine learning algorithms. The ultimate task of the agent is to maximize the reward, making the setting of the reward reflect the understanding of the algorithm designer for the solution of the problem and its desired goal. For example, Aslani et al. used queue length as an action reward [33]. Mannion et al. used the waiting time of vehicles as a reward function [34] and Casas [29], Van Der Pol et al. [31] used speed as a reward function. Some scholars use indicators, such as the number of stops, road network throughput, and intersection pressure (the non-equilibrium of traffic flow distribution at intersections), to represent the rewards of signal strategies.

Finally, the structure of a deep neural network has an important influence on a RL algorithm. The core problem of traffic signal control is to establish the mapping between traffic state characteristics and signal phase, with different scholars deploying different neural network structures to process the input data. Wade et al. [25] used a convolutional neural network topology to process DTSE state data, Choe et al. [21] used a circular neural network to process traffic data with time-series data, and Li et al. [32] used an auto-encoder for traffic state feature selection of high-dimensional input data, removing the noise in the traffic data.

2.1 | PROPOSED METHOD

2.1.1 | Problem statement

The paper aims to control a typical urban road intersection with traffic signals. The signal control of intersection is mainly for motor vehicles, and it does not consider the setting of pedestrian exclusive phase since the pedestrian crossing behaviour is often accompanied by the movement of a vehicle. As shown in Figure 1, there is a left turn lane, a straight lane, and a right turn lane in the direction of each intersection entrance. At the intersections, each traffic signal phase gives priority to a group of non-collision vehicle directions. According to the real-time traffic conditions, the system uses an adaptive way to select the signal phase in the next time interval. This system aims to reduce the queue length and the waiting time by maximizing the average vehicle speed to guide vehicles through the intersection more effectively. This control process can be defined as a RL problem.

2.2 | RL definition

Three important factors need to be defined in the training process of reinforcement learning: state, action, and reward, called the $< S, A, R >$ status. It is difficult to directly capture the intersection top view picture and the corresponding coding because the DTSC method is easy to be limited by the field environment in operation. For this reason, this paper uses the feature-based value vector state representation method. Here, we divide each entrance direction into three lanes: left, straight, and right lane, respectively, and five characteristic indexes of traffic volume, average vehicle speed, average queue length, average waiting time, and lane occupancy are used as the characteristic expression of traffic state. Since traffic data possesses time-series characteristics, it is necessary to add the time dimension in state representation. In this paper, the traffic states with seven decision steps in history are selected. Therefore, a traffic state $s(s \in S)$ at an intersection can be represented by a vector of 12∗5∗7.

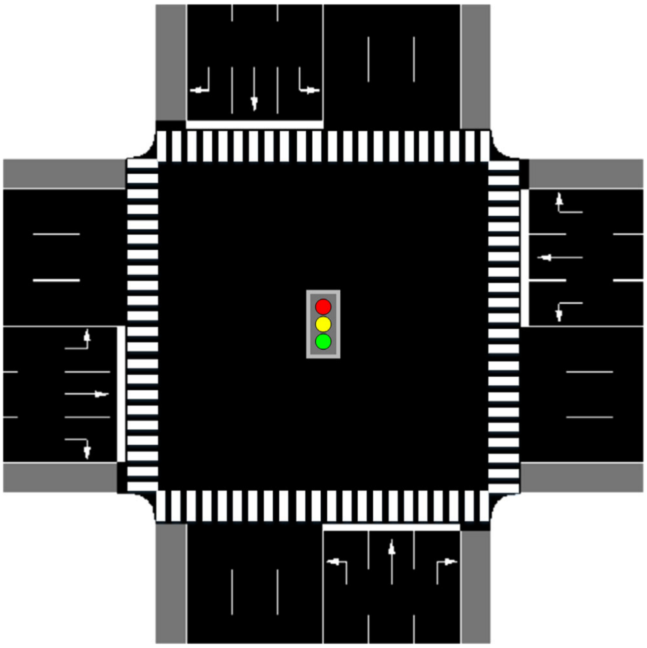


FIGURE 1 A signal-controlled standard intersection

The action $a(a \in A)$, in a signal control problem, is the phase of the traffic signal. Under the premise that right turns are not controlled, and considering the common signal phasing at actual intersections (one-way traffic is relatively rare and usually inefficient), we only allow simultaneous traffic in two directions, and the action space of the algorithm is reduced accordingly. Based on this, we remove potentially conflicting signal phases, such as eastbound straight and westbound left turn, to obtain all feasible eight conflicting signal phases. As a result, the action selection space in this paper is set to eight green signal phases. They are the NSS, the EWS, the NSL, the EWL, the east-west straight and left turn (ESL), the West Straight and left turn (WSL), the South straight and left turn (SSL), and the NSL, as shown in Figure 2. In this system, the right turn direction is unified without control and is always set in the release state. The agent can only select to execute one of the eight defined green signal phases in each time point, without allowing multiple phases to be executed at the same time.

The role of reward $r(r \in R)$ in reinforcement learning is to provide immediate feedback regarding previously selected actions. In the setting of traffic signal control, the commonly used evaluation criteria include queue length, waiting time, average speed, and others. Therefore, this paper selects the average speed of road network vehicles as the reward value of environmental feedback. The training objective of reinforcement

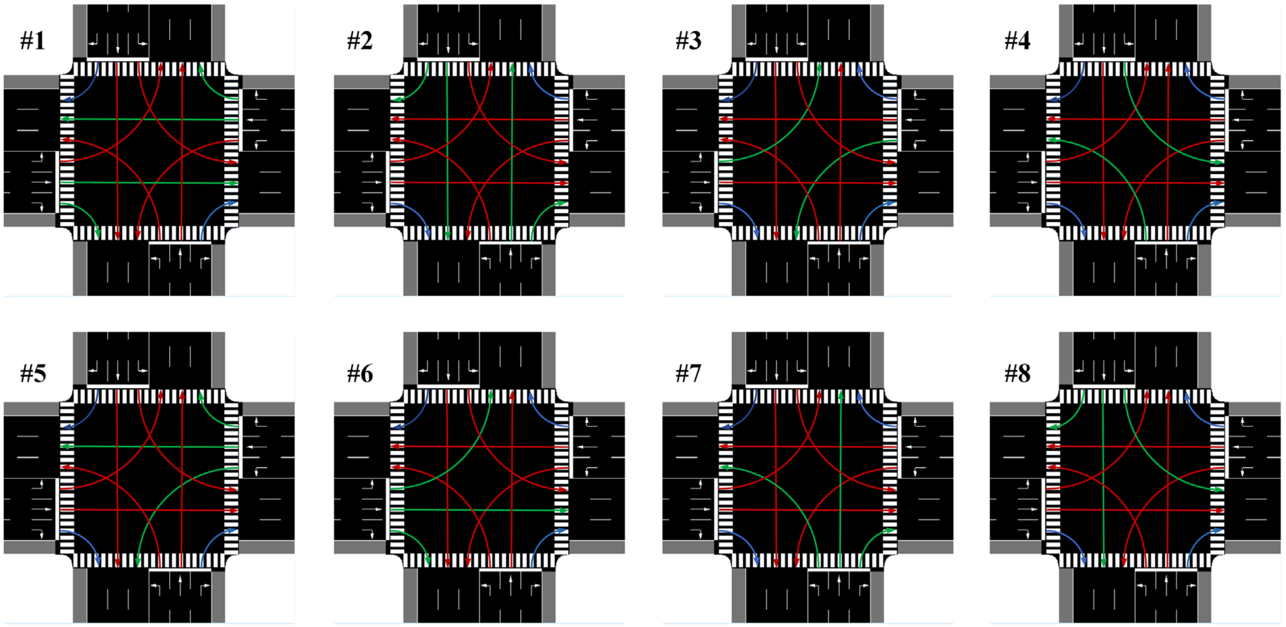


FIGURE 2 All signal phases in the action space

learning is to maximize the cumulative reward, and particularly to maximize the average vehicle speed. In the evaluation period, the higher the average speed of vehicles at each time, the shorter the time of traffic flow through the intersection, and the higher the traffic efficiency. Thus, the average speed of vehicles can directly reflect the advantages and disadvantages of the signal control phase.

2.3 | Agent

In reinforcement learning, agents learn the best signal control strategy through their interaction with the traffic environment. In this paper, the agent that controls the signal phase is modelled using optimal PPO. LSTM can extract discriminant information from the state space and get the optimal action strategy through value evaluation using a well-designed LSTM as an approximator of the value function and strategy function, avoiding in such way the curse of dimension brought by the huge state-action space. The goal of reinforcement learning is to maximize the cumulative expected reward under the control of the strategy function:

$$RL \text{ target} : \max \bar{R}_\theta = \max E_{\tau \sim p_\theta(\tau)}[R(\tau)] = \max \sum_\tau p_\theta(\tau) R(\tau) \tag{1}$$

where τ represents the trajectory data during the interaction between the agent and the environment, $\tau = \{s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_T, a_T, r_T\}$, θ represents the parameter of the policy function in the agent, $R(\tau)$ represents the turn cumulative reward of the track τ, and $p_\theta(\tau)$ represents the probability of a trajectory. The parameter update direction of the policy function should be made consistent with the gradient direction of the expected reward to maximize the expected reward. Suppose the data of N rounds is sampled and the gradient ascent method is used to update the parameters of the strategy function:

$$\begin{aligned} \theta &\leftarrow \theta + \eta \nabla_\theta \bar{R}_\theta \\ \nabla_\theta \bar{R}_\theta &= E_{\tau \sim p_\theta(\tau)}[R(\tau) \nabla_\theta \log p_\theta(\tau)] \\ &= \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \nabla_\theta \log p_\theta(a_t^n \,|\, s_t^n) R(\tau^n) \end{aligned} \tag{2}$$

where η denotes the learning rate, $\nabla_\theta \overline{R}_\theta$ represents the gradient direction of the policy function update, and $p_\theta(a_t^n / s_t^n)$ indicates the probability that the output of the strategy function is a_t when the input of the nth round is s_t.

The classical policy gradient model uses an on-policy method to update the parameters. Every time the parameters of the policy function are updated, the data needs to be collected again for the next update of the parameters. Therefore, the efficiency of samples usage is low and the training time is long. Moreover, utilizing the accumulated reward as the reward of each action weakens the actions with high reward values, partially reduces the probability of good action, and increases the probability of low reward action.

Therefore, the PPO algorithm first uses off-policy to update parameters to improve the training efficiency and accurately evaluate the value of movement. Furthermore, two sets of strategy function parameters are set for sampling and training, respectively. The strategy function π_θ, which is used for sampling, continuously interacts with the traffic environment, while the output signal controls the phase. π_θ does not update the parameters before the predefined number of interaction rounds is reached. The strategy function π_θ, which is used for training, is deployed to adjust the gradient direction and continuously update parameters and does not participate in the environmental interaction process. When the predefined number of rounds is reached, the parameter π_θ is replaced with π_θ. At the same time, to ensure the consistency, continuity, and stability of model parameter updating, the importance sampling method is introduced as shown in the following equation:

$$E_{\tau\sim p_\theta(\tau)}[R(\tau)\nabla_\theta \log p_\theta(\tau)]$$

$$= E_{\tau\sim p_{\theta'}(\tau)}[\frac{p_\theta(\tau)}{p_{\theta'}(\tau)}R(\tau)\nabla_\theta \log p_\theta(\tau)] \tag{3}$$

where $p_\theta(\tau)$ represents the probability of the trajectory τ under the control of the strategy function π_θ. Thus, when $p_\theta(\tau)$ is close to $p_{\theta'}(\tau)$, the action output probability of the strategy function π_θ, which is used for training, is close to the one of the strategy function π_θ, which is used for sampling. This can implement the training mode conversion to the off-policy one.

Consequently, the reward of each action should be redefined in the next turn. When calculating the reward of the current action, it is necessary to add the reward of the next time point to the current reward applying a certain discount ratio as the action reward value. Subtracting a benchmark reward value from the discounted action reward value to get the final reward of the current action, as defined in (4), is required to avoid the situation that the reward value is always positive. The benchmark reward value is the output using the value function based on the LSTM.

$$R(\tau) \leftarrow A^{\theta'}(s_t, a_t) = \sum_{t'=t}^{T_n} \gamma^{t'-t} r_{t'}^{n} - V^{\pi_v}(s_t) \tag{4}$$

where $A^{\theta'}(s_t, a_t)$ is the advantage function, taking action a_t, B the difference between the discount reward and the average action reward expectation under the state of s_t, and γ represents the future discount factor $\lambda < 1$. The value function π_v is the estimation of the average action reward expectation of the value function $V^{\pi_v}(s_t)$ in the state s_t.

Finally, the penalty terms of π_θ' and π_θ distribution divergence are introduced to effectively control the influence of the distribution difference between the sampling strategy function and the training strategy function on the parameter updating, and the gradient direction of the policy function π_θ update is defined as

$$\nabla_\theta J^{\theta'}(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t\,|s_t)}{p_{\theta'}(a_t\,|s_t)}A^{\theta'}(s_t,a_t)\nabla_\theta \log p_\theta(a_t\,|s_t)\right]$$
$$-\beta\nabla_\theta KL(\theta,\theta')$$
$$KL(\theta,\theta') = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} p_\theta(a_t\,|s_t)\log\frac{p_\theta(a_t\,|s_t)}{p_{\theta'}(a_t\,|s_t)} \tag{5}$$

where $J^{\theta'}(\theta)$ is the objective function of parameter updating, β is the influence weight of the distribution divergence of the strategy function π_θ', and $\pi_\theta KL(\theta,\theta')$ is the divergence of the distribution of the strategy functions π_θ' and π_θ.

Another important aspect is the selection of the neural network structure. The long-term and short-term memory networks, proposed by Hochreiter and Schmidhub [23], are widely used to solve time-series problems with high dependency. Therefore, this paper deploys an LSTM network as the main structure of decision function and value function to establish an effective mapping relationship between the traffic state characteristics and the signal phase decision. The internal structure and calculation flow of the LSTM network are shown in Figure 3 and as defined in the following formula:

$$\begin{cases} f_t = \sigma(W_f\cdot[h_{t-1},x_t] + b_f) \\ i_t = \sigma(W_i\cdot[h_{t-1},x_t] + b_i) \\ \tilde{C}_t = \tanh(W_C\cdot[h_{t-1},x_t] + b_C) \\ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t = \sigma(W_o\cdot[h_{t-1},x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{cases} \tag{6}$$

where $x_t \in R^d$ represents the input vector of the LSTM unit, $f_t \in R^h$ represents the activation vector of the forgetting gate, $i_t \in R^h$ represents the activation vector of the input gate, $C_t^{\%} \in R^h$ represents the candidate cell state vector, $C_t \in R^h$ represents the cell state vector, $o_t \in R^h$ represents the activation vector of the output gate, $h_t \in R^h$ represents the hidden state vector, also called as LSTM, $W = i^{h\times h}$ denotes the output vector of the unit, and $b \in R^{h\times h}$ represents the weight matrix and deviation vector parameters that need to be learned in training, with the superscripts d and h denoting the number of input features and the number of hidden cells, respectively. σ is the sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, tanh is the hyperbolic tangent function, $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

The processing process of the LSTM-PPO model is shown in Figure 4 and the complete process can be described as follows. The current state s is first fed into an LSTM-based Actor-New model, which is used to construct the action distribution. The action sampled from the distribution is then input to the environment to obtain the next state s_and the reward r. At the same time, the set of state, action, and reward will be stored as a sample. The new state s_ will be considered the current state to repeat the above environmental interaction process, which will loop a certain number of times until the sample size reaches a predetermined number. At the end of the loop, the state s_of the last step will be fed into the Critic-NN model to get value v_, and then the discount reward will be calculated to update v_. All the states s in the stored samples will be fed into the Critic-NN model to get the corresponding value v. Subsequently, the difference between v and v_, that is, advantage, is then used to calculate *c_loss* and thus update the parameters of the Critic-NN model. All the states s will also be fed into both Actor-New and

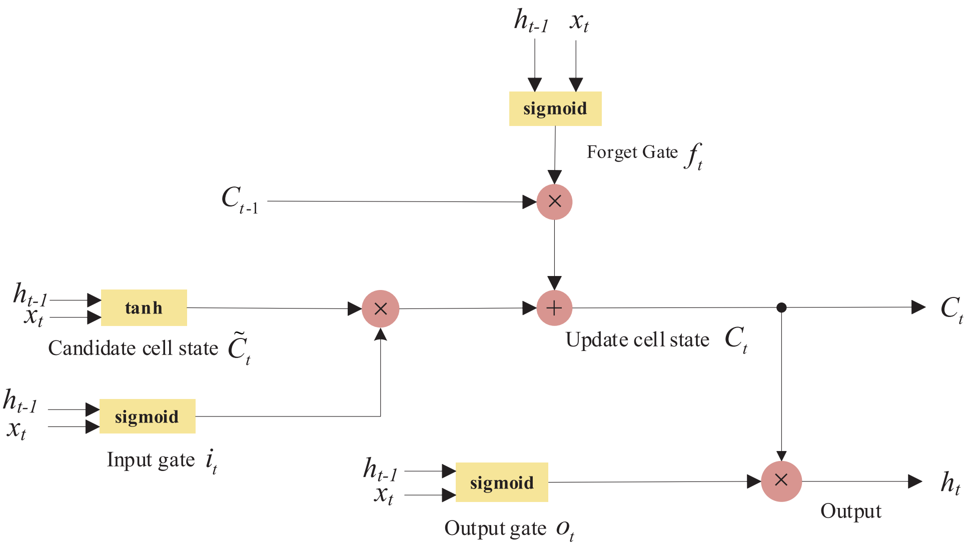


FIGURE 3 LSTM network architecture. LSTM, long short-term memory.

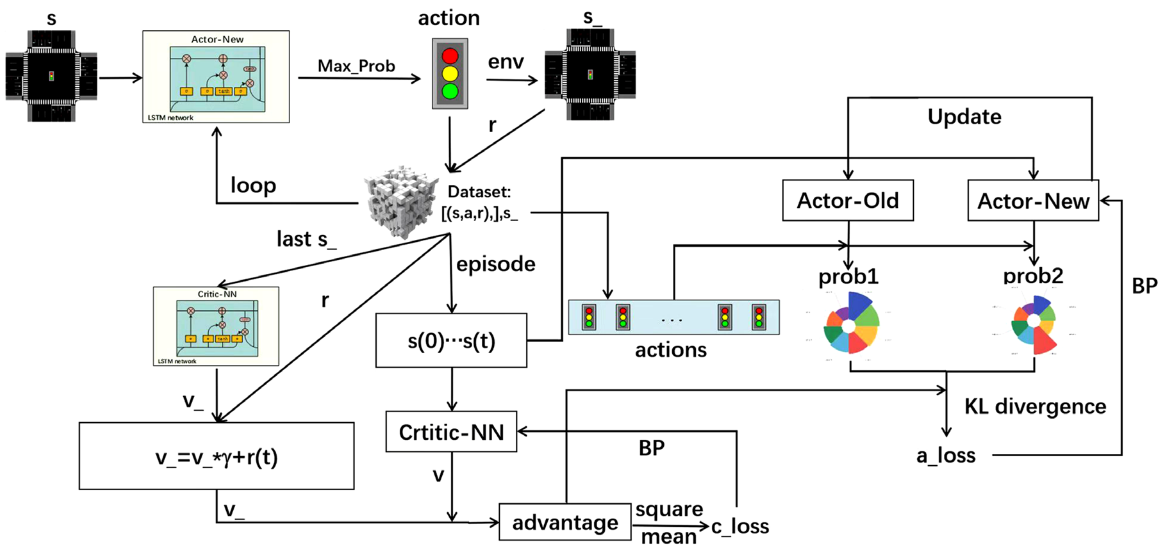


FIGURE 4 Processing flow of LSTM-PPO model. PPO, proximal policy optimization.

Actor-Old model, where the latter has the same network structure as Actor-New. The obtained probabilities prob1 and prob2 will be used to calculate the *KL* divergence and thus update the Actor-New model. After a certain number of iterations, the weights of Actor-New will be used to update the Actor-Old model. At this point, the process of updating the parameters of the Actor and Critic models is completed, and the algorithm repeats the above process so as to continuously optimize.

3 | NUMERICAL EXPERIMENTS

3.1 | Simulation environment

All the experiments in this paper are carried out on the traffic micro simulator simulation of urban mobility (SUMO) to verify the effectiveness of the model. SUMO provides an application programming interface implemented in Python programming language, through which user-defined functions can be implemented for traffic simulations.

As illustrated in Figure 1, the signalized intersection is implemented in SUMO. The application programming interface (API) of SUMO is used to set the vehicle size, vehicle driving control, signal phase control, path selection, and detector. The car-following model and the lane-changing model used in the SUMO simulation are the default models of the simulation system, in which the car-following model car following-Krauss is modified from the model proposed by Stefan Krauß [36] to increase the driving speed as much as possible on the basis of ensuring safety and solving the potential collision problem. The lane-changing model LC2013 [37], developed by Jakob Erdmann, clearly distinguishes four different lane change motives and proposes corresponding lane change strategies for different lane change scenarios. To

TABLE 1 Simulation settings

Parameter name	Parameter value
The length of the driveway	500 m
Vehicle length	5 m
Minimum safe distance between vehicles	2.5 m
Maximum vehicle speed	16.67 m/s (60 km/h)
Maximum vehicle acceleration	2.6 m/s^2
Maximum vehicle deceleration	4.5 m/s^2
Transition phase duration	3 s
Signal phase duration	30 s
Simulation time step	0.1 s
Path selection	Free choice, can be changed midway
Vehicle input	3600 pcu/h
E2 detector length	250 m
E1 detector position	5 m before the stop line

TABLE 2 Hyper-parameter settings for the training step of the proposed method

Hyper-parameter name	Hyper-parameter value
Number of epochs	700
Initial learning rate	0.0005
Batch size	32
Number of units in the LSTM layer	64
Number of units in fully connected layers	200, 8
Number of historical time points	7
Signal-phase decision interval	30 s
Maximum training sample capacity	1200
Number of sampled training samples	300
Number of iterations in each training	100

simulate the real traffic flow as realistically as possible, we approximate the traffic flow according to the Poisson distribution, where vehicles are randomly generated and distributed throughout the road network, and are introduced at the same time interval. The detailed setting parameters are summarized in Table 1.

The real-time control of the signal phase can be achieved through SUMO's TraCI module. In this experiment, the basic duration of the signal phase is set to 30 s. Meanwhile, this article first presets 300 standard passenger cars in the simulation environment of the intersection to simulate the running state of the intersection during peak hours. Then different locations of the intersection and input standard passenger cars to the intersection at a speed of 3600 vehicles per hour are randomly selected. In the meantime, modelling assumes that the simulated vehicle can freely choose the path, but the moving distance is at least 500 m to ensure that enough vehicles are entering and passing through the intersection.

3.2 | Hyper-parameters

The simulation experiment runs on a workstation with a processor Intel core R Xeon (R) ES-2650 v4 @2.20 GHz 32G CPU and a graphics card 2 NVIDIA TITAN Xp 24G GPUs. The main training parameters of the LSTM-PPO single-point signal control model proposed in this paper are inferred by a thorough trial and error process. Both the strategy network (Actor-Old/New) and the evaluation network (Critic-NN) use a neural network structure with an LSTM layer (units = 64), a fully connected layer (units = 200), and a fully connected layer (units = 8) of three layers. The dimensions of the input matrix are set as [7, 12, 5], with 7 being the historical time-points, 12 being the intersection directions, and 5 the traffic state indicators (traffic volume, average vehicle speed, average queue length, average parking waiting time, average lane occupancy). The detector collects traffic state data according to the signal phase decision interval (30 s), and the reward function of the model is set as the average speed of all vehicles in the road network calculated by the SUMO simulation system.

The total number of iterative simulation rounds of the model was 700 epochs and the duration of each round was 30 min. Moreover, the signal phase needs to be switched 60 times per round. Three hundred data samples are collected every time the training sample capacity stored by the model exceeds 1200. The strategy network and the estimation network are trained iteratively 100 times, the number of samples sent to training in each batch is 32, and the optimizer selects the adaptive moment estimation optimization. The initial training learning rate for both networks is set to 0.0005. The settings of all hyper-parameters are shown in Table 2. The rest of the comparison algorithms involved in the experiment in this chapter, except for the differences in the neural network structure, the settings of the training parameters, and the settings of the simulated environment, were consistent to ensure the credibility of the experimental comparison results.1

3.3 | Training method

3.4 | Results and discussion

Seven hundred rounds of iterative simulation simulations were carried out in this paper to verify the adaptive ability of the LSTM-PPO single-point signal control model. The changes in the five traffic state indicators of the average traffic volume, the average vehicle speed, the average queue length, the average parking waiting time, and the average lane occupancy during the entire learning process of the model were monitored. The control effect of the LSTM PPO single-point signal control model was compared with an adaptive algorithm SCOOT and one of

ALGORITHM 1 LSTM-PPO based single-point signal control algorithm

```
Input: Maximum number of steps $S_{\max}$, Actor-New network $A_{\text{new}}$, Actor-Old network$A_{old}$, Critic network $C$, Number of actor network updates $N_A$, Number of critic network updates $N_C$, Replay Memory $M$Output: The optimal Actor-New network $A_{\text{new}}$ and Critic network $C$

Initialize replay memory with capacity $L$
Initialize the parameter of $A_{\text{new}}$ and $A_{\text{old}}$
for each simulation do
    initialize $s$ with current view of the intersection
    repeat # each time step in the simulation
        choose action $a$ based on the output distribution of the $A_{\text{new}}$
        take action $a$, observe reward$r$and next state $s\_$
        store transition ($s$, $a$, $r$, $s\_$) in $M$
    until reach $S_{\max}$
    obtain value from the Critic network $v\_=C(s\_)$
    calculate discounted reward $r_t$ for each time step $t$
    obtain advantage value $A_t$ from Critic network $A_t = C(s_t, r_t)$ for each time step $t$
    if the replay memory size exceeds $L$ then
        train and update the parameters of $A_{\text{new}}$ for $N_A$ times
        train and update the parameters of $C$ for $N_C$ times
        replace the parameter of $A_{\text{new}}$ with $A_{\text{old}}$
        clear replay memory $M$
    end if
```

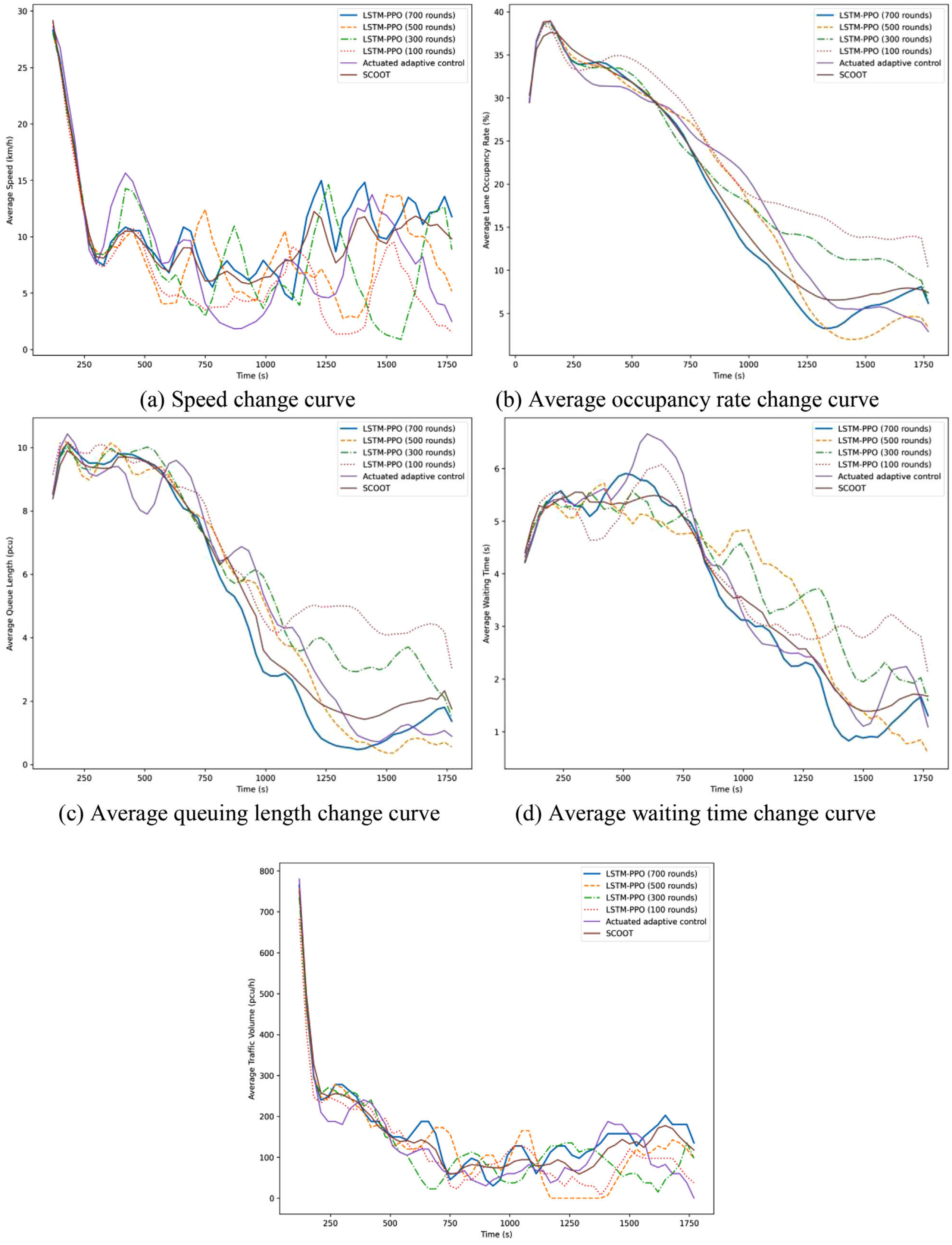
the built-in actuated algorithms in the SUMO simulation software, and the results are shown in Table 3. The Actuated Traffic Lights method built in the SUMO simulation software is the method performing actuated control based on the time interval among them. When the electromagnetic coil induces continuous traffic flow, which happens when the time interval between vehicles is less than the maximum gap setting value, the existing green light phase time is extended and otherwise, it switches to the next phase. At the same time, the detector and stop the following parameters could be set: line time distance, maximum phase duration, dynamic phase selection, and others. SCOOT is a real-time, online, dynamic, and stable traffic system that continuously monitors the traffic demand in the approach lanes of each intersection of the road network, so as to optimize the signal timing scheme at each intersection and keep the dynamics of each intersection in a state with the least delay time and number of stops. Its related control strategy is implemented through mathematical model simulation, which can tolerate partial error detection information, and can quickly adapt to complex and changing traffic conditions.

In general, under the control of the newly introduced model, the average speed and the average traffic flow have been greatly improved, while on parallel the average lane occupancy rate, the average queue length, and the average waiting time were decreased in different magnitudes, and the operation efficiency of the intersection has been in a state of continuous improvement. Specifically, after 700 rounds of simulation training, the average running speed of vehicles at the intersection gradually increased from 8.232 to 11.944 km/h, which corresponds to an increase of 45.09%. The average lane occupancy rate, the average queue length, and the average parking waiting time were decreased by 21.38%, 25.86%, and 12.94%, respectively, while the average traffic volume was increased by 24.55%. It is obvious from the changing of the trend of these five evaluation indicators that the operating efficiency of the traffic flow at intersections has been significantly improved, showing that with the increase of the number of simulation rounds, the LSTM-PPO model adaptively adjusts the signal strategy output through iterative learning to identify the signal control phase that makes the traffic state continuously improving. It is also noteworthy that from the comparison of the SCOOT algorithm implemented in SUMO, with the LSTM-PPO model the latter has increased by 5.4% in speed and 6.8% in traffic volume after 700 rounds of training, and the other three indicators have also decreased by about 5% to 9%. This proves that the LSTM-PPO signal control model can exceed the classic signal control method through continuous training.

In addition, in terms of calculation speed, the calculation time of each signal phase inference of the LSTM-PPO model is maintained at 2.3 milliseconds and the average training time is about 0.86 min per round in the 700 rounds of simulation training, indicating thus that the parameter changes during the model training process are relatively stable. As for the decision time of the proposed algorithm, although it is higher than that of the adaptive control algorithm, it can still adequately meet the practical requirements.

Next, the LSTM-PPO model was evaluated with the number of training iteration taking values 100, 300, 500, 700, keeping the actuated traffic lights control in the same simulation environment, simulating the 30-min intersection operation, and observing the traffic status under the control of different models. As shown in Figure 5, when comparing the control effects of each model, the LSTM-PPO model with 700 rounds of iterative simulation training outperformed the other models. The three indicators of the average queue length, the average parking waiting time, and the average lane occupancy rate have maintained their lowest state since the middle of the simulation. Although the speed change of the vehicle is oscillating, it can be seen that the area enclosed by the speed curve and the time axis of the proposed model are the largest, which shows that the running distance obtained by the integral of the speed curve along the time dimension is the largest one. At the same time, comparing the changes in traffic volume, it can be concluded that under the control of the LSTM-PPO model of 700 rounds of iterative simulation training, the vehicle can pass through the intersection at a faster speed.

A variety of basic reinforcement learning methods are also used to compare the results with the model introduced in this paper to verify the performance of the proposed model more comprehensively. The deployed reinforcement learning methods included the reinforcement learning algorithm DQN based on the value function learning and its improved algorithms Double DQN (DDQN), and the Double Dueling



(a) Speed change curve

(b) Average occupancy rate change curve

(c) Average queuing length change curve

(d) Average waiting time change curve

(e) Average traffic volume change curve

FIGURE 5 The process of traffic state change under the control of different stages

TABLE 3 LSTM-PPO model training effect comparison

Parameter name	100 rounds	300 rounds	500 rounds	700 rounds	Actuated adaptive control	SCOOT
Average speed (km/h)	8.232	9.945	10.029	11.944	10.101	11.328
Average share (%)	23.956	21.624	19.392	18.834	20.196	19.977
Average queue length (pcu)	6.608	6.017	5.166	4.899	5.354	5.312
Average waiting time (s)	4.042	4.012	3.767	3.519	3.802	3.685
Average traffic (pcu/h)	169.831	178.475	182.542	211.525	177.458	197.885
Average training time (min)	0.862	0.865	0.861	0.863	—	—
Average decision time (ms)	2.319	2.285	2.307	2.301	1.586	1.845

DQN (3DQN), as well as the PPO algorithm using DTSE proposed by Ma et al. [35]. The five indicators (i.e. average speed, average land occupancy rate, average maximum queue length, average parking time, and the traffic flow) are compared in Figure 6.

In terms of average speed, the LSTM-PPO single-point signal control model has the highest average vehicle speed, and the 3DQN model has the worst effect. With the increase of iteration rounds, the speeds of DQN and DDQN were both raised slowly, but the speed fluctuations were relatively large, and the speed curve was not stable, showing that the update of the model parameters only depends on the evaluation of the action value, overestimate phenomenon will be observed. The adjustment of the model's parameter is in accordance with the direction of the expected maximum reward in the future. If an action is sampled multiple times, then its reward estimate will be amplified again and again, but in reality, the actual reward may not be so high, or even be in the opposite state, introducing thus prediction errors. In such a case instability will be observed in the training process. In contrast, the PPO (DTSE) and the LSTM-PPO models quickly increased the average running speed of vehicles, and they maintained a slow upward trend.

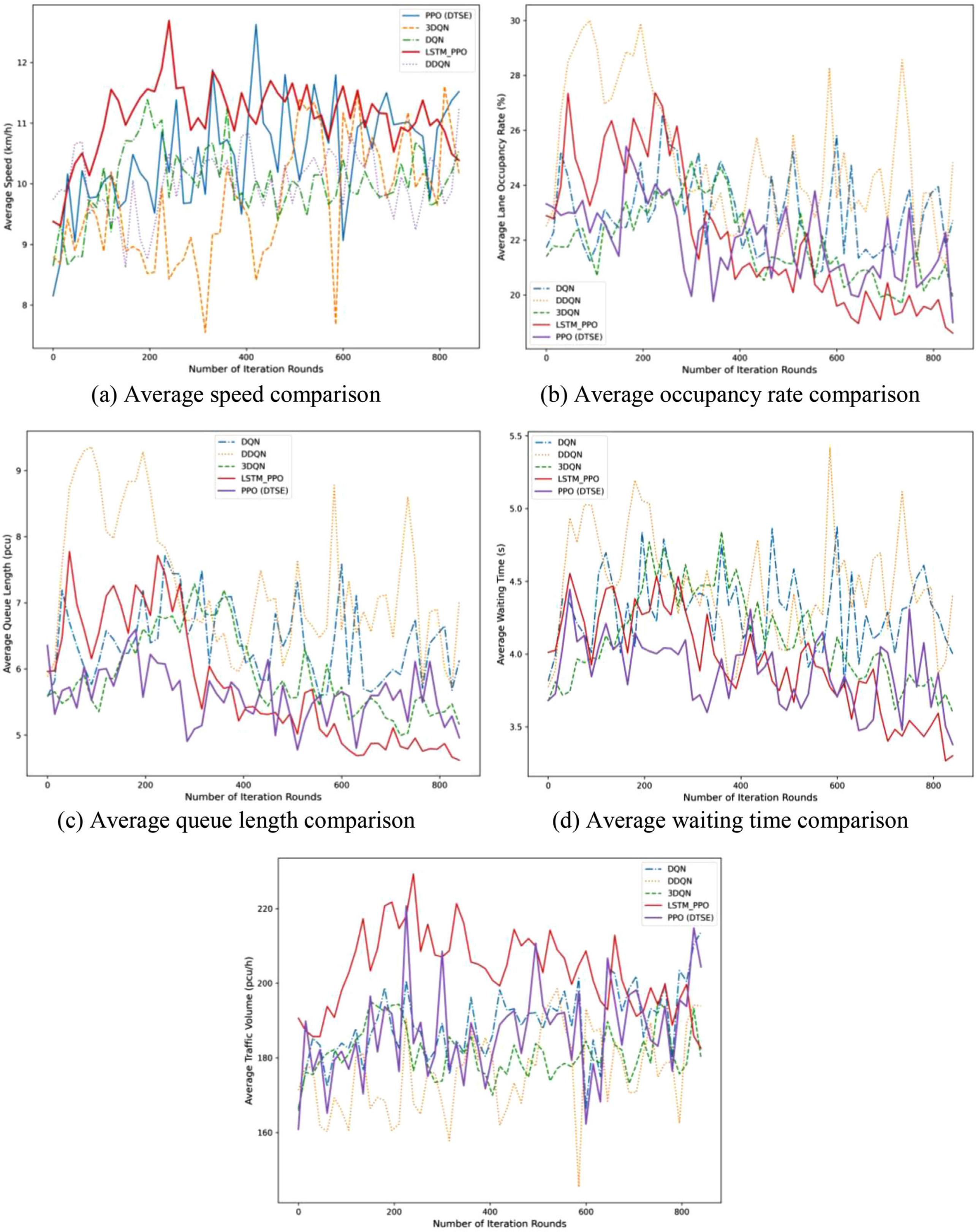
The average lane occupancy rate, the average maximum queue length, and the average parking time of the LSTM-PPO single-point signal control model were the largest in terms of descent speed and amplitude. However, from the perspective of average parking time, each model encountered a training bottleneck when it was close to 500 rounds of iterations, and the LSTM-PPO model effectively adjusted the signal control strategy after the parking time curve rose slightly, and suppressed the average parking of vehicles in the average vehicle waiting time. In addition, it is worthwhile noting that the PPO (DTSE) model usually performs better in the initial training phase.

The analysis of traffic flow can intuitively reflect the operating efficiency of the intersection. In a limited time interval, when the intersection is close to saturation, the signal control using the proposed method could always keep the traffic at the intersection at the highest position, while the operating efficiency was higher than the one other model.

4 | CONCLUSION

This paper introduces the LSTM-PPO single-point signal control model, which uses the long- and short-term memory network LSTM to extract the time-series characteristics of the traffic state, and optimizes the PPO algorithm through the near-end strategy. The iterative training of the adaptive intelligent strategy adjusted the signal control strategy. This paper used the simulation environment constructed by SUMO and compared it with the LSTM-PPO model to verify the model's control performance and self-learning and self-adaptive capabilities, using five metrics: average speed, average lane occupancy, average queue length, average parking time, and traffic flow. The control effects of the various iterative versions, including adaptive signal control methods, classic reinforcement learning control algorithms DQN, DDQN, 3DQN, and PPO (DTSE), and of the detailed analysis of the signal phase selection method of the LSTM-PPO model were measured and compared. The experimental results revealed that the control effect of the LSTM-PPO model proposed in this paper significantly outperformed other models, indicating that the modelling of the time dependence of traffic data is helpful for the selection of signal control strategies, and it was proved that the reinforcement learning algorithm of the near-end strategy optimization allows the model to possess self-learning and adaptive intelligent control capabilities, which can, in turn, optimize the control strategy and realize the improvement of the operating efficiency of the intersection.

However, the LSTM-PPO model still has room for further improvement in terms of convergence and stability. In the case of single intersection signal control, the proposed model only needs to sense the change of traffic flow state at the intersection, while in the case of multi-intersection signal control, a multi-agent signal control system needs to be established. In order to realize signal phase coordination between intersections, it is necessary to further improve the expression of traffic state space and realize multi-scale perception of traffic flow characteristics. At the same time, it is necessary to adjust the reward function, including global and local situation assessment, to ensure coordination among multiple agents. The application of this model to signal control of multiple intersections is the difficulty



(a) Average speed comparison

(b) Average occupancy rate comparison

(c) Average queue length comparison

(d) Average waiting time comparison

(e) Average traffic volume comparison

FIGURE 6 Comparison of the control effects of the five models

and focus of the following research. Moreover, when applying reinforcement learning methods to actual intersection control, factors such as pedestrians, non-motorized vehicles, and road conditions may need to be taken into consideration to improve the robustness of the model.

AUTHOR CONTRIBUTIONS

Liben Huang: Data curation; Investigation; Methodology; Software; Validation; Writing – original draft. Xiaohui Qu: Conceptualization; Funding acquisition; Supervision; Writing – review & editing

ACKNOWLEDGEMENTS

This research was funded by the Industrial Chain Demonstration Application Scenario Construction Project of Nanjing Industrial and Informatization Development Special Funds in 2020 (Grant Nos. 012947218/2020-133671), Soft Science Research Project of Nanjing Association for Science and Technology in 2021, and supported by Nanjing Social Science Foundation Project.

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. Zhu, L., Yu, F.R., Wang, Y., Ning, B., Tang, T.: Big data analytics in intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst. 20(1), 383–398 (2018)
2. Traffic signal settings. [cited 01 Jul 2021]. https://trid.trb.org/view/113579 (1958).
3. Guo, M., Wang, P., Chan, C.Y., Askary, S.: A reinforcement learning approach for intelligent traffic signal control at urban intersections. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 Oct, pp. 4242—4247. IEEE, Manhattan, NY, USA(2019). https://doi.org/10.1109/ITSC.2019.8917268.
4. Roess, R.P., Prassas, E.S., McShane, W.R.: Traffic Engineering, 4th ed. Pearson/Prentice Hall, Hoboken, NJ (2011)
5. Little, J.D., Kelson, M.D., Gartner, N.H.: MAXBAND: A versatile program for setting signals on arteries and triangular networks. 1981 Jan [cited 01 Jul 2021], pp. 1–30. https://dspace.mit.edu/bitstream/handle/1721.1/1979/SWP-1185-08951478.pdf?sequence=1.
6. Varaiya, P.: The max-pressure controller for arbitrary networks of signalized intersections. In: Ukkusuri, S.V., Ozbay, K. (eds.). Advances in Dynamic Network Modeling in Complex Transportation Systems, pp. 27–66. Springer, Heidelberg, Germany (2013)
7. Cools, S.B., Gershenson, C., D'Hooghe, B.: Self-organizing traffic lights: A realistic simulation. In: Prokopenko, M. (ed.) Advances in Applied Self-Organizing Systems, pp. 45–55. Springer, Heidelberg, Germany (2006)
8. Vincent, R., Peirce, J.: 'MOVA': Traffic responsive, self-optimising signal control for isolated intersections. TRRL Research Report. 1988, 01 Jul 2021. https://trid.trb.org/view/295257.
9. Kronborg, P., Davidsson, F.M.: LHOVRA: Traffic signal control for isolated intersections. Traffic Eng. Control. 34(4), 195–200 (1993)
10. Kronborg, P., Davidsson, F.: Development and field trials of the new SOS algorithm for optimising signal control at isolated intersections. In: Eighth International Conference on Road Traffic Monitoring and Control, London, UK, 23–25 Apr, pp. 80–84. IET Digital Library, Stevenage, UK (1996). https://doi.org/10.1049/cp:19960295.
11. Park, B.B., Myzie, C., Agbolosu-Amison, S.J.: Improving actuated traffic signal control operations using concept of dynamic Gap-out. In: Ninth International Conference on Applications of Advanced Technology in Transportation (AATT), Chicago, IL, 13–16 Aug, pp. 617–622. ASCE, Reston, VA, USA (2006). https://doi.org/10.1061/40799(213)98.
12. Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., et al: Learning phase competition for traffic signal control. In: 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 Nov, pp. 1963—1972 ACM Digital Library, New York, NY, USA (2019). https://doi.org/10.1145/3357384.3357900.
13. Sims, A.G., Dobinson, K.W.: The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. IEEE Trans. Veh. Technol. 29(2), 130–137 (1980)
14. Hunt, P.B., Robertson, D.I., Bretherton, R.D., Royle, M.C.: The SCOOT on-line traffic signal optimisation technique. Traffic Eng. Control. 23(4), 190–192 (1982)
15. Prabuchandran, K., HK, A.N., Bhatnagar, S.: Multi-agent reinforcement learning for traffic signal control. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 Oct, pp. 2529–2534. IEEE, Manhattan, NY, USA (2014). https://doi.org/10.1109/ITSC.2014.6958095.
16. Shabestary, S.M.A., Abdulhai, B.: Deep learning vs. discrete reinforcement learning for adaptive traffic signal control. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 Nov, pp. 286–293. IEEE, Manhattan, NY, USA (2018). https://doi.org/10.1109/ITSC.2018.8569549.
17. Thorpe, T.L., Anderson, C.W.: Traffic light control using SARSA with three state representations. 1996 [cited 01 Jul 2021]. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.5406&rep=rep1&type=pdf.
18. Mannion, P., Duggan, J., Howley, E.: An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control. Springer International Publishing, Berlin (2016)
19. Abdoos, M., Mozayani, N., Bazzan, A.: Traffic light control in nonstationary environments based on multi agent Q-learning. In: International IEEE Conference on Intelligent Transportation Systems. IEEE (2011)
20. Liang, X., Du, X., Wang, G., Han, Z.: Deep reinforcement learning for traffic light control in vehicular networks. arXiv e-prints. 29 Mar 2018 [cited 01 Jul 2021]. https://ui.adsabs.harvard.edu/abs/2018arXiv180311115L.
21. Choe, C.J., Baek, S., Woon, B., Kong, S.H.: Deep Q-learning with LSTM for traffic light control. In: 2018 24th Asia-Pacific Conference on Communications (APCC), Ningbo, China, 12–14 November, pp. 331–336. IEEE, Manhattan, NY (2018). https://doi.org/10.1109/APCC.2018.8633520.
22. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv e-prints. 20 Jul 2017 [cited 01 Jul 2021]. https://ui.adsabs.harvard.edu/abs/2017arXiv170706347S.
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997)
24. Mousavi, S.S., Schukat, M., Howley, E.: Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intell. Transp. Syst. 11(7), 417–423 (2017)
25. Genders, W., Razavi, S.: Using a deep reinforcement learning agent for traffic signal control. arXiv e-prints. 03 Nov 2016 [cited 01 Jul 2021]. https://ui.adsabs.harvard.edu/abs/2016arXiv161101142G.
26. Shi, S., Chen, F.: Deep recurrent Q-learning method for area traffic coordination control. J. Adv. Math. Comp. Sci. 27(3), 1–11 (2018)
27. Garg, D., Chli, M., Vogiatzis, G.: Deep reinforcement learning for autonomous traffic light control. In: 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 3–5 Sep, pp. 214–218. IEEE, Manhattan, NY, USA (2018)
28. Jang, I., Kim, D., Lee, D., Son, Y.: An agent-based simulation modeling with deep reinforcement learning for smart traffic signal control. In: 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 17–19 Oct, pp. 1028–1030. IEEE, Manhattan, NY, USA. (2018). https://doi.org/10.1109/ICTC.2018.8539377.

29. Casas, N.: Deep deterministic policy gradient for urban traffic light control. arXiv e-prints. 27 Mar 2017 [cited 01 Jul 2021]. https://ui.adsabs.harvard.edu/abs/2017arXiv170309035C.
30. Ge, H., Song, Y., Wu, C., Ren, J., Tan, G.: Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control. IEEE Access 7, 40797–40809 (2019)
31. Van Der Pol, E.: Deep reinforcement learning for coordination in traffic light control. Master Dissertation, University of Amsterdam, North Holland, Netherlands (2016)
32. Li, L., Lv, Y., Wang, F.Y.: Traffic signal timing via deep reinforcement learning. IEEE/CAA J Automatic Sinica. 3(3), 247–254 (2016)
33. Aslani, M., Mesgari, M.S., Wiering, M.: Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. Transp. Res. Part C Emerg. Technol. 85, 732–752 (2017)
34. Mannion, P., Duggan, J., Howley, E.: An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In: McCluskey, T., Kotsialos, A., Müller, J., Klügl, F., Rana, O., Schumann, R. (eds.) Autonomic Road Transport Support Systems, pp. 47–66. Birkhäuser, Switzerland (2016)
35. Ma, Z., Cui, T., Deng, W., Jiang, F., & Zhang, L.: Adaptive Optimization of Traffic Signal Timing via Deep Reinforcement Learning. Journal of Advanced Transportation. Hindawi (2021)
36. Krauss, S.: Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. Ph.D thesis, Hauptabteilung Mobilitt und Systemtechnik des DLR Koln, Germany (1998)
37. Erdmann, J.: SUMO's lane-changing model. In: Lecture Notes in Control and Information Sciences, 13, Seiten, pp. 105–123. Springer Verlag. 2nd SUMO User Conference, 2014, Berlin (2015)

How to cite this article: Huang, L., Qu, X.: Improving traffic signal control operations using proximal policy optimization. IET Intell. Transp. Syst. 17, 592–605 (2023). https://doi.org/10.1049/itr2.12286