OXFORD

Transportation Safety and Environment, 2022, **4**: tdac027

DOI: 10.1093/tse/tdac027

Research Article

Traffic signal control in mixed traffic environment based on advance decision and reinforcement learning

Yu Du[1,2], Wei ShangGuan[1,2,3,*] and Linguo Chai[1]

[1]School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China;
[2]The State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China;
[3]Beijing Engineering Research Centre of EMC and GNSS Technology for Rail Transportation, Beijing Jiaotong University, Beijing 100044, China.
*Corresponding author. E-mail: wshg@bjtu.edu.cn

Abstract

Reinforcement learning-based traffic signal control systems (RLTSC) can enhance dynamic adaptability, save vehicle travelling time and promote intersection capacity. However, the existing RLTSC methods do not consider the driver's response time requirement, so the systems often face efficiency limitations and implementation difficulties. We propose the advance decision-making reinforcement learning traffic signal control (AD-RLTSC) algorithm to improve traffic efficiency while ensuring safety in mixed traffic environment. First, the relationship between the intersection perception range and the signal control period is established and the trust region state (TRS) is proposed. Then, the scalable state matrix is dynamically adjusted to decide the future signal light status. The decision will be displayed to the human-driven vehicles (HDVs) through the bi-countdown timer mechanism and sent to the nearby connected automated vehicles (CAVs) using the wireless network rather than be executed immediately. HDVs and CAVs optimize the driving speed based on the remaining green (or red) time. Besides, the Double Dueling Deep Q-learning Network algorithm is used for reinforcement learning training; a standardized reward is proposed to enhance the performance of intersection control and prioritized experience replay is adopted to improve sample utilization. The experimental results on vehicle micro-behaviour and traffic macro-efficiency showed that the proposed AD-RLTSC algorithm can simultaneously improve both traffic efficiency and traffic flow stability.

Keywords: Adaptive traffic signal control, mixed traffic flow control, advance decision-making reinforcement learning

1. Introduction

Intelligent intersection control (IIC) is a primary research area within the field of the intelligent transportation system. The key idea behind IIC is to apply optimal intersection passing rules that utilize real-time traffic status information. With the development of technologies such as autonomous driving, the urban traffic environment will gradually form a mixed traffic environment. A mixed traffic environment consists of numerous traffic participants with differing intelligent levels, such as human-driven vehicles (HDVs) and connected automated vehicles (CAVs), fixed time traffic signal lights and adaptive traffic signal lights. Developing intersection optimization methods in mixed traffic environments will be the long-term trend of IIC.

Adaptive traffic signal control systems can 'decide' to change the signal according to intersection sensing data, such as the waiting length. However, the unpredictability of signal changes can lead to dangerous intersection situations. The need to improve signal control efficiency and predictability for drivers in mixed traffic environments makes the development of intersection control methods a priority. In recent years, the proliferation of artificial intelligence has brought new possibilities for the development of intelligent transportation systems [1, 2]. For example, the self-driving vehicle is expected to bring dramatic changes in terms of energy consumption, safety, access and time savings [3–5]. Motivated by the limitations of current intersection control systems, the aim of the present paper is to propose a self-learning signal control method that addresses three key challenges:

1). Take advantage of emerging technologies such as reinforcement learning (RL) to improve intersection control performance for mixed traffic environments.

2). Overcome the shortcomings of adaptive signal control, which are unpredictable and often lead to dangerous behaviour at intersections.

3). Design a decentralized intersection control algorithm that can be extended to a widely road network.

The use of reinforcement learning (RL) in the traffic signal control problem has gained significant interests [6–11]. The model-free RL method Q-learning is frequently used to approximate the optimal solution in traffic signal control. In most early studies, low-complexity traffic features were used as the input [12]. The experience replay and target network technologies are involved to automatically extract useful features (machine-crafted features) from raw traffic data [13]. In Ref. [14], the state was defined as vehicle location and vehicle speed, and the reward was the difference in cumulative vehicle waiting time between signal cycles.

With the development of reinforcement learning methods, there is widespread interest in optimizing decision making and improving training speed. Xu et al. [15] proposed a targeted transfer reinforcement learning based method which uses unsuper-

Received: December 30, 2021. **Revised:** March 27, 2022. **Accepted:** April 17, 2022

vised networks to evaluate the similarity of samples, and thus accelerate the learning process. In Ref. [16], three input states from low to high resolution were extracted from traffic data and their performance comparisons were derived by an actor critic algorithm. The experimental results revealed that machine learning methods are affected by the accuracy of traffic information. From the existing work, there is consensus that the system design approach significantly impacts on the performance of machine learning-based traffic signal controllers.

The contributions of this paper are threefold.

First, we propose a method to train the reinforcement learning-based traffic signal control (RLTSC) using scalable traffic information. The sensing range of the intersection is modified dynamically to improve data processing efficiency. The coverage of 5G communication technology that could be applied in IIC in the future is a few hundred metres [17]. In contrast to previous work, which only provided the designing of action and reward functions of their reinforcement learning algorithm, the relationship between the traffic status sensing range and the traffic light control period is proposed through theoretical analyses and simulation experiments. Based on our analyses, we propose a Region-based RLTSC (Region-RLTSC) method, which can train an optimal control policy for isolated intersections.

Second, the unpredictability of adaptive traffic signal control (TSC) is solved to achieve smooth control in mixed traffic flow. Most previous research on RLTSC did not consider the human factor, such as reaction times and comfortable acceleration for drivers. With existing methods, the vehicle cannot predict signal light changes, which can cause the vehicle to become involved in dangerous situations, such as sudden braking. We propose the advance decision-making method that provides all vehicles with the next state of traffic lights ahead of time, and thereby avoids dangerous behaviour at intersections.

Third, the impact of the proportion of intelligent traffic signal lights on large-scale traffic networks is explored. Our experiment exams the influence of agents on mixed traffic flow, two aspects are considered: *i*) the proportion of intelligent traffic signal controller, and *ii*) the penetration rate of CAVs.

The following sections of this paper are organized as follows. Section 2 introduces our proposed Region-RLTSC strategy. Two new concepts are proposed to optimize the design of algorithm components, trust region state and standardized reward. In addition, the cycle of action is analysed to match the travelling demand. In Section 3, based on the Region-RLTSC, the advance decision-making reinforcement learning algorithm (AD-RLTSC) is proposed and discussed to solve the unpredictability of traditional adaptive traffic signal control methods. An introduction on how to train the control policy by using the Double Duelling Deep Q-learning Network (3DQN) algorithm is presented in Section 4. In Section 5, we verify our algorithm by simulations and compare its performance to popular traffic signal control algorithms. In Section 6 we present our conclusions.

2. Region-based reinforcement learning traffic signal control

The traffic signal light system can be considered as an intelligent agent, which can observe the states from the intersection and return a phase selection action, as shown in Fig. 1. Our goal is to fit an optimized signal timing policy to make the agent take the optimal action. Modelling the states, actions and rewards are the critical issues in designing an RLTSC system. This section introduces the proposed Region-RLTSC algorithm.

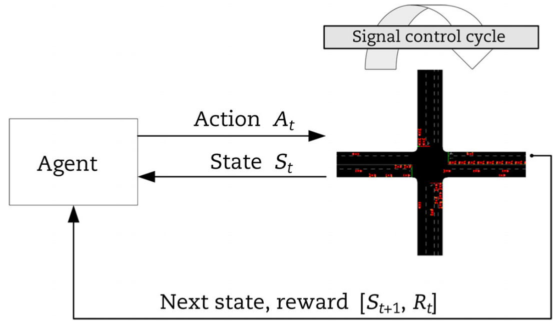

Fig. 1. Schematic diagram of the traditional RLTSC model.

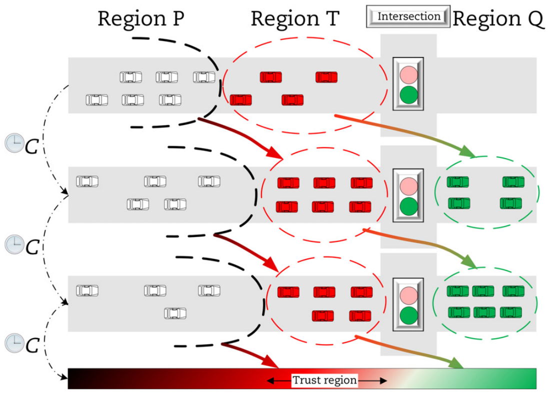

Fig. 2. Schematic diagram of the proposed trust region.

2.1 Input state for advance decision making

An adequate sensing distance is crucial in the learning process of RL based traffic signal control method. Limited vehicles' information is not enough for agent's decision making. On the contrary, excessive information can exacerbate instability and make neural network training difficult.

Fig. 2 is a schematic diagram of the perception region. In one control period, the vehicles in Region T will pass the intersection, the vehicles in Region P cannot pass and the vehicles in Region Q do not affect the intersection. Therefore, we define the distance of Region T as the trust region.

The perception range, denoted as d_p, represents the length of the trust region and refers to the traffic data perception range, which is calculated as Eq. (1). The control cycle refer to the interval of time between two actions.

$$d_p = C \times \bar{v} \tag{1}$$

where, C is the control circle and v is the average speed. Fig. 3 is a schematic diagram to explain the implementation of the trust region state. The information within the d_p is helpful for decision-making. However, a dynamically changing matrix cannot be fed into a neural network as the input matrix. Hence, for implementation, the trust region state $\mathbf{S}_t$ is defined as an element-wise multiplication of a binary mask matrix $\boldsymbol{M}$ and the full-scale state matrix $\mathbf{S}$, as shown in Eq. (2). In Refs. [12–13], the grid technology is employed to define the state matrix, and we observe two types of data, vehicles' position and speed. There are three lanes in each direction and four directions in total. Therefore, the size of the full-scale state matrix $\mathbf{S}$ and the trust region state $\mathbf{S}_t$ are $L \times 12 \times 2$, where L is as defined as Eq. (3).

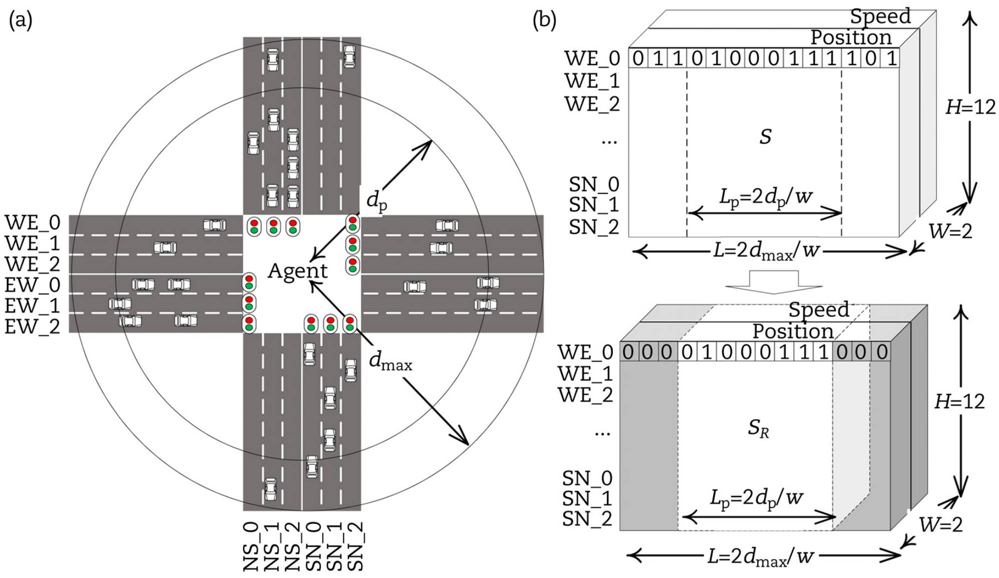

Fig. 3. Schematic diagram of intersection model and trust region state matrix: (a) perception range; (b) input matrix.

$$\mathbf{S}_t = \mathbf{S} \times \mathbf{M} \tag{2}$$

$$L = \left\lfloor \frac{2d_{\max}}{w} \right\rfloor \tag{3}$$

$$M_{i,j} = \begin{cases} 1, \text{if } i, j \in \left[\frac{d_{\max}-d_p}{w}, \frac{d_{\max}+d_p}{w}\right] \\ 0, \textit{else} \end{cases} \tag{4}$$

$$d_{\max} = C \times v_f \tag{5}$$

The mask matrix $\boldsymbol{M}$ is defined as Eq. (4), where $d_{\max}$ is the largest perception distance can be provided, v_f is the free flow speed and w is the size of the grid. The function of the mask matrix $\boldsymbol{M}$ is to select the perception range d_p. Only vehicles that satisfy this distance constraint can be observed by the agent.

2.2 Standardized reward

The reward function reflects the expected benefit of the actions. The goal of the TSC is to keep the traffic flow efficient and smooth. Therefore, two indicators, the queue length and vehicle speed, are involved in our reward function [18]. It should be noted that the transportation system is a highly dynamic system. Congestion at intersections is closely related to the traffic demand from upstream. Because of real-time changes in traffic demand it is possible to get different temporal rewards even if the agent chooses the same action.

Therefore, a standardized reward function is adopted in this paper. At time t, the reward for the agent is defined as Eq. (6). At the end of the control cycle, the reward is the average of cumulative rewards over a control cycle, defined as Eq. (7).

$$R_t = \alpha \times \tanh\left(\overline{Q_b} - Q_t\right) - (1-\alpha) \times \tanh\left(\overline{V_b} - V_t\right) \tag{6}$$

$$R_m = \frac{\sum_{t=0}^{C} R_t}{C} \tag{7}$$

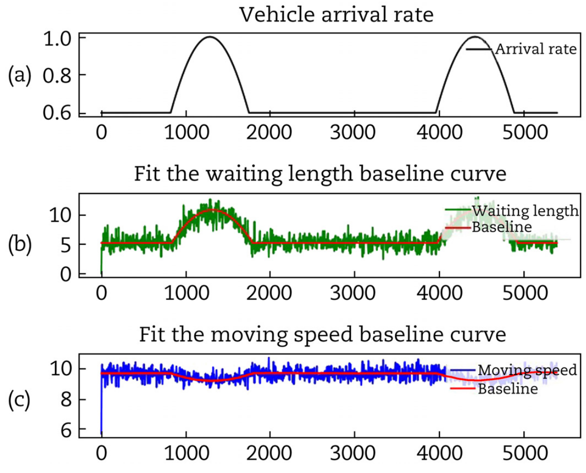

Fig. 4. The baselines of the two factors in the reward function are tested by the fixed time control method: (a) input arrival rate; (b) baseline of the waiting length; (c) baseline of the average speed. (The red solid line is the baseline function fitted by the least square method.)

In Eq. (6), two indicators are involved in the reward value, namely the average queue length and vehicle speed respectively. $\overline{Q_b}$ and $\overline{V_b}$ are the baseline values under the current vehicle arrival rate, respectively. Q_t and V_t are the real-time feedback values at time t. To simultaneously satisfy the two goals of maximizing vehicle speed and minimizing queue length, the function tanh is used to scale the two factors to a uniform range. α is invoked as an impact factor and is set as 0.5 through experiments.

The baseline values are acquired from the fixed time control method, as in Fig. 4. According to the traffic flow steady-state theory, the delay time and stopping rate of vehicles at the signal control intersection mainly depends on the vehicle arrival rate and the capacity of the intersection [19]. However, when the vehicle arrival rate is dynamically changing, queue waiting length and average vehicle speed depend on not only the current traffic volume saturation but also the previous traffic conditions. To gain the baseline values under the assumption of dynamic traffic volume and to eliminate the deviation caused by the modelling of traffic simulation software, we tested the fixed time control method with the same control cycle and the same traffic volume via traffic simulations.

It can be seen from Eq. (6) that if the agent acts better than the fixed time controller, it will get a positive reward; otherwise, it will

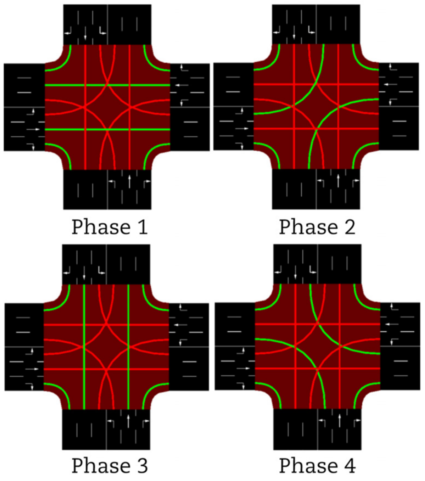

Fig. 5. The signal phases represented by four actions.

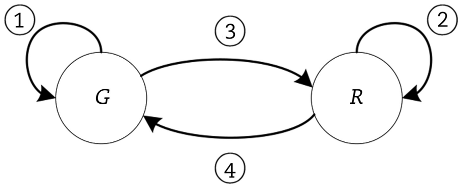

Fig. 6. Schematic diagram of transition relationships.

get a negative punishment. If the agent acts as well as the fixed time controller, the reward will be 0.

2.3 Action

Consider an isolated intersection, which has two straight and two left-turn phases, as illustrated in Fig. 5. The 'action' of the agent is to choose one phase among all four phases in every control cycle.

The critical issue of action modelling is how to choose a suitable signal control cycle, denoted as C earlier in this article. In accordance with the traffic control theory, the minimum period time which is the total time of all phases is defined as:

$$C_{\min} = \frac{N \times l}{1 - \left(\frac{q_c}{3600/h_s}\right)} \tag{8}$$

where, N denotes the number of phases in one period; l is to the activation time of one phase; q_c refers to the sum of traffic volume in key lanes; h_s indicates the saturation headway. The equation suggests that the traffic demand is proportional to the minimum period required. Given the analysis in Section 2.1, the signal control cycle is proportional to the perception distance. Accordingly, the minimum control period is expanded as traffic demand rises and a higher traffic demand requires a larger perception distance. The design of the control cycle should satisfy the traffic demand. A short control cycle and a small perception range are applicable for the intersection with small traffic demands, while the intersection with large traffic demands requires a long control cycle as well as a large perception range.

In brief, the control cycle should match the traffic demand at the intersection and should be minimized from the point of view of efficiency. Accordingly, the minimum acceptable control period is adopted as the control cycle of the agent.

Table 1. The relationship between the decision light state and the light display state.

Transition between two decision states	Traffic light display state
1 (green → green)	Green
2 (red → red)	Red
3 (green → red)	Green countdown
4 (red → green)	Red countdown

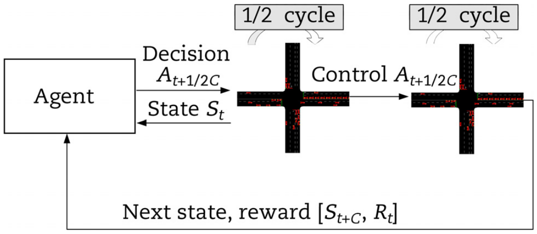

Fig. 7. Schematic diagram of transition relationships.

3. Advance decision-making reinforcement learning traffic signal control

In existing studies, the TSC agent will obtain the current state and take another phase at the end of a control cycle. However, there is uncertainty about the next phase and the vehicle does not know whether the current phase will change. The uncertainty can lead to dangerous behaviours (e.g. sudden braking at intersections). To address this problem, we propose the advance decision-making reinforcement learning algorithm for traffic signal control and a novel traffic signal mechanism, termed as a bi-countdown timer.

3.1 Advance decision-making reinforcement learning

The core idea of the Advance Decision-making Reinforcement Learning (ADRL) control algorithm is to separate the decision cycle from the control cycle and generate the action of the following control cycle in advance.

The decision cycle and the control cycle have the same cycle interval C, and the decision time is earlier than the control time. To simplify the later analysis, we set the decision cycle half a cycle in advance of the control cycle. At the decision time t, the traffic information within the trust region d_p is captured by the intersection information sensing sensor, such as a camera, and stored as S_t. The state matrix $\mathbf{S}_t$ is fed to the deep reinforcement learning network as the input. The optimal action A_t serves as output which is the next activated phase. Each lane direction is controlled by an independent signal light, including four states: green, red, green countdown and red countdown. The decision of each lane has two states: green and red. At the beginning of the decision cycle, the decision light state is compared with the current light state. For a lane, if the decision state is different from the current state, the current light will count down from 1/2C to 0, and the decision light state will be activated when the countdown is finished. Otherwise, if the decision state is same as the current state, the current light will remain until the beginning of the next decision cycle. The output decision has two states: green and red, as shown in Fig. 6. The two adjacent decision states constitute four kinds of transition relationships. Four kinds of transition relationships determine the traffic light display state: green, red, green countdown and red countdown, as in Table 1.

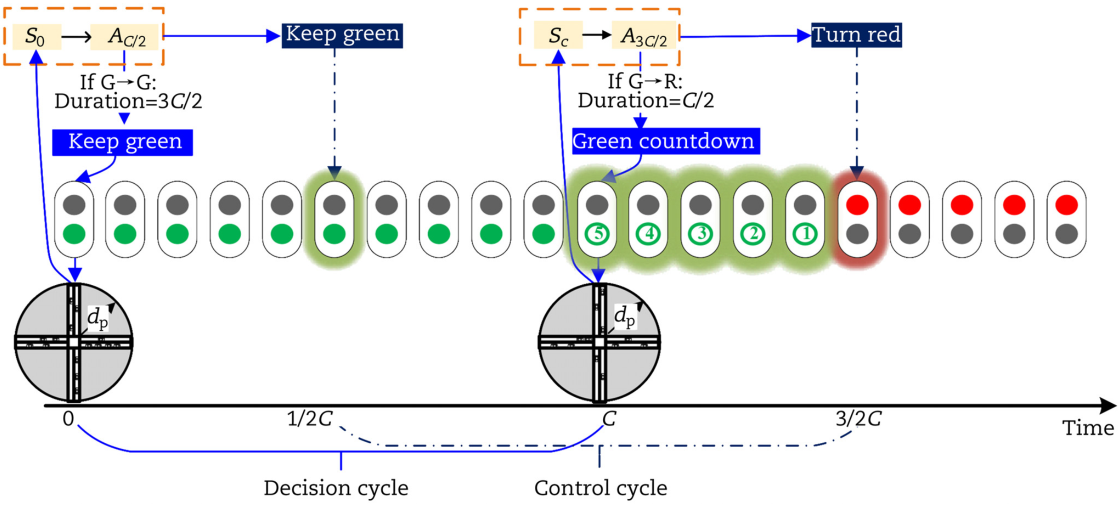

Fig. 8. An example for AD-RLTSC.

To separate the decision and control process, the state matrix is adjusted. The output action, which is calculated at the decisive moment t, will be executed at future time $t + 1/2C$. Therefore, the input for the decision process will be the state in the near future. We use $\mathbf{S}_{t+1/2C}$ instead of the current state $\mathbf{S}_t$ to achieve the advance decision. There are two ways to get the state matrix $\mathbf{S}_{t+1/2C}$. The first is to capture the state matrix $\mathbf{S}_t$ and predict $\mathbf{S}_{t+1/2C}$ according to $\mathbf{S}_t$. However, the prediction process is a difficult task and will introduce new systematic errors. From the analysis in the trust region state, we could adjust the intersection sensing range instead of predicting the state matrix. Then at the decision moment, the critical vehicles will be the vehicles which pass through the intersection at time [1/2C, 1/2C + C], and the speed range of vehicles is $[v_{\min},\ \bar{v}]$. Considering that the current speed of the vehicles queued at the intersection, and thus the minimum vehicle speed is set to 0. Finally, the perception range d_p is adjusted to Eq. (9).

$$d_p = \frac{3}{2} \times C \times \bar{v} \tag{9}$$

Fig. 7 shows the model of our proposed ADRL algorithm. In general, it incorporates two adjustments compared to the traditional reinforcement learning algorithms.

First, we separate the agent's 'action' to two processes: 'decision' and 'control'. At the intervals during which the decision has been made but the agent has not been controlled, a reminder signal is sent to all vehicles in the mixed traffic environment.

Second, we make future action decisions by expanding the sensing range of the state matrix, utilizing the relationship between the sensing range and the control period.

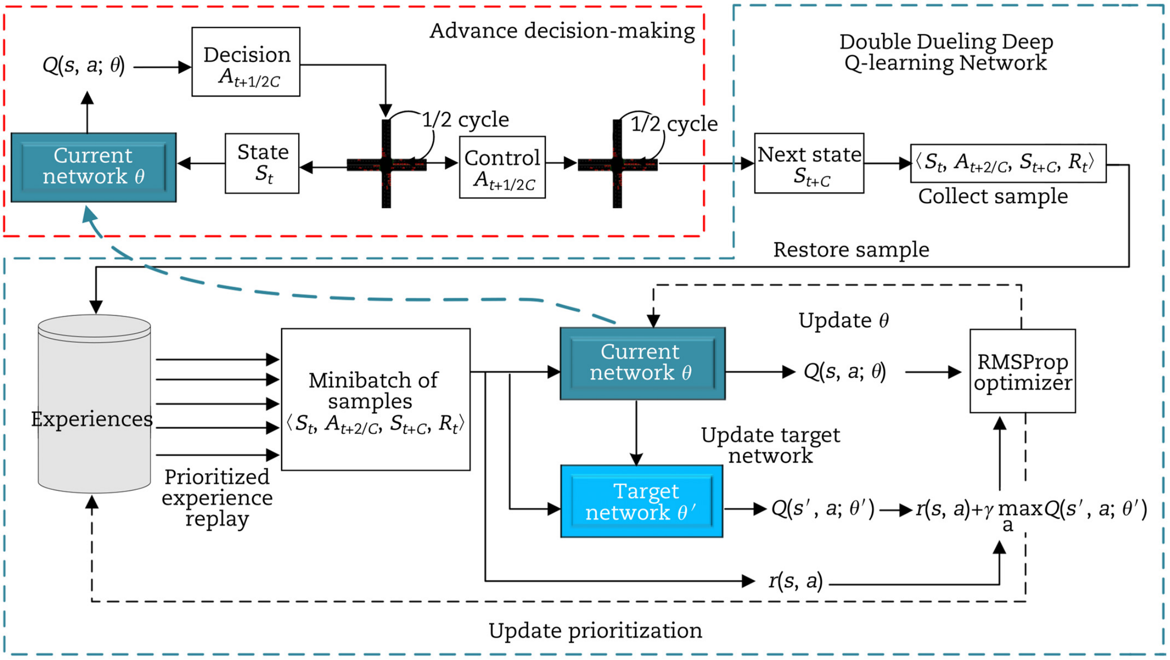

Fig. 9. The overall architecture of the AD-RLTSC algorithm.

3.2 Bi-countdown timer

Generally, a traffic light has three display states, which are red, green and yellow. Some cities in China (e.g. Nanjing) have started to use countdown timers to eliminate safety hazards at intersections.

The proposed bi-countdown timer consists of two hold states and two countdown states. The two hold states refer to green and red, suggesting that the current state will be kept more than time 1/2C. The two countdown states reveal that the signal state will change at the end of the countdown.

The benefits of the bi-countdown timer and AD-RLTSC cover three points:

1) For the human driver, it acts as a acceptable signal rule. The proposed Bi-countdown timer can provide the precise remaining time of the current phase. The state of the light can vary from red to green or from green to red.

2) For the intelligent vehicles, the remaining signal time can be exploited to optimize driving speed. Since the next state of the traffic light is hard to predict, almost all existing adaptive traffic light control methods are not capable of providing the information about phase change prediction. Using the AD-RLTSC control method, driving speed control methods can be adopted to vehicles with differing intelligent levels.

3) Bi-countdown timer is capable of saving phase loss time and achieving short-cycle control at intersections. It can achieve a long switching reminder time without taking up green time. Accordingly, the control period can be very short to enhance intersection efficiency.

Fig. 8 presents an example to illustrate the AD-RLTSC algorithm. At the beginning of the decision cycle, *time* = 0, the agent captures the current state matrix $\mathbf{S}_0$ and makes the decision of light state at time C/2. The output decision $A_{C/2}$ is 'green', the same as the current light state; thus, the traffic light will stay green for one more cycle. At C, the first decision cycle ends and the second decision cycle starts. The state matrix $\mathbf{S}_c$ is obtained and the output decision light state $A_{3C/2}$ is 'red', which differs from the current display state of light. Subsequently, the light display state will change from 'green' to 'red' at the beginning of the control cycle at 3C/2. During the time [C, 3C/2], the green countdown timer will start and count from C/2 to 0. At 3C/2, the decision $A_{3C/2}$ will be executed. The light state turns into 'red' and holds at least to 2C.

4. 3DQN

The 3DQN algorithm is used for the policy function approximation. The network's input is the observed intersection state and the output is a vector of estimated Q-values. Prioritized experience replay is employed to improve the training efficiency.

4.1 Double Deep Q-Network

Q-learning is a model free learning algorithm, attempting to evaluate how good it is to use current policy to act *a* in state *s* using the Q-value. The Q-value is defined as Eq. (10). The solution of Q-value can be formalized as Eq. (11), where $r(s,\ a)$ is the immediate reward received, $\max Q(s,\ a)$ is the highest possible Q-value from state *s* and γ is a discount factor. The strategy takes the action that maximizes the Q-value to get the highest reward in the long term.

$$Q\left(s,\ a\right) = E\left(\sum_{k=0}^{\infty} \gamma^k R_{t+k} \middle| \mathbf{S}_t = s,\ \ A_t = a, \pi\right) \tag{10}$$

$$Q\left(s,\ a\right) := r\left(s, a\right) + \gamma \max_a Q\left(s',\ a\right) \tag{11}$$

The deep Q-Network (DQN) was proposed by DeepMind in 2015 [20]. The idea of DQN is to approximate the Q-value using neural network. Then the Q-value is denoted as $Q\ (s,\ a;\ \theta)$, where θ represents the weights of the network. The training aims to minimize the difference between the current Q-value, $Q(s,\ a;\ \theta)$, and the target Q-value. The target Q-value is the sum of immediate and future rewards, as Eq. (12).

$$L = \left[Q\left(s,\ a;\ \theta\right) - \left(r\left(s,\ a\right) + \gamma \max_a Q\left(s',\ a;\ \theta\right)\right)\right]^2 \tag{12}$$

In DQN, both true label (target Q-value) and samples (current Q-values) are generated from the same neural network. To eliminate the impact of changing goals, the double Q-learning algorithm is employed [21]. In the double DQN, a target network is adopted to generate the target Q-value, while the current action is generated from the current network. The cost of the training is expressed in Eq. (13).

$$L = \left[Q\left(s,\ a;\ \theta\right) - \left(r\left(s,\ a\right) + \gamma \max_a Q\left(s',\ a; \theta'\right)\right)\right]^2 \tag{13}$$

4.2 The duelling network architecture and prioritized experience replay

The duelling network architecture replaced the single-stream Q network with two streams, representing the state values and (state-dependent) action advantages, respectively. Intuitively, the duelling architecture can estimate which states are (or are not) valuable, without learning the effect of each action for each state. The modified Q function is expressed in Eq. (14), where V denotes the value function and A represents the advantage function [22].

$$Q\left(s,\ a;\ \theta,\ \alpha, \beta\right) = V\left(s;\ \theta, \beta\right) + A\left(s,\ a;\ \theta, \alpha\right) \tag{14}$$

If the experience inputs are uniformly sampled from a replay memory, the probability of the occurrence of samples will be ignored, making a few valuable samples hard to apply for training. Prioritized experience replay was proposed in 2015 and widely used to improve the performance of the DQN [23]. In TSC, the sample of the rush hour is more important than the other time. Prioritized experience replay is involved for the following three reasons. First, increasing the probability that important samples in rush hours to be sampled; second, forgetting previous experiences to be avoided; third, correlations between experiences to be reduced. The experiences are ranked as the value of δ, which represents the temporal difference error, as shown in Eq. (15). p_i is the priority of the experience *i*, which is the reciprocal of the rank value, as shown in Eq. (16). P_i is the final sampling probability, and τ denotes the importance factor, as defined in Eq. (17).

$$\delta_i = \left|Q(s,\ a;\ \theta)_i - Q(s,\ a;\ \theta')_i\right| \tag{15}$$

$$p_i = \frac{1}{\delta_i} \tag{16}$$

$$P_i = \frac{p_i^\tau}{\sum_k p_k^\tau} \tag{17}$$

4.3 Learning architecture

Fig. 9 shows the overall architecture of the proposed AD-RLTSC algorithm. There are mainly two parts in the overall architecture, advance decision-making traffic signal control and the training process of reinforcement learning network. The input state is obtained from the traffic condition in real time. And the action

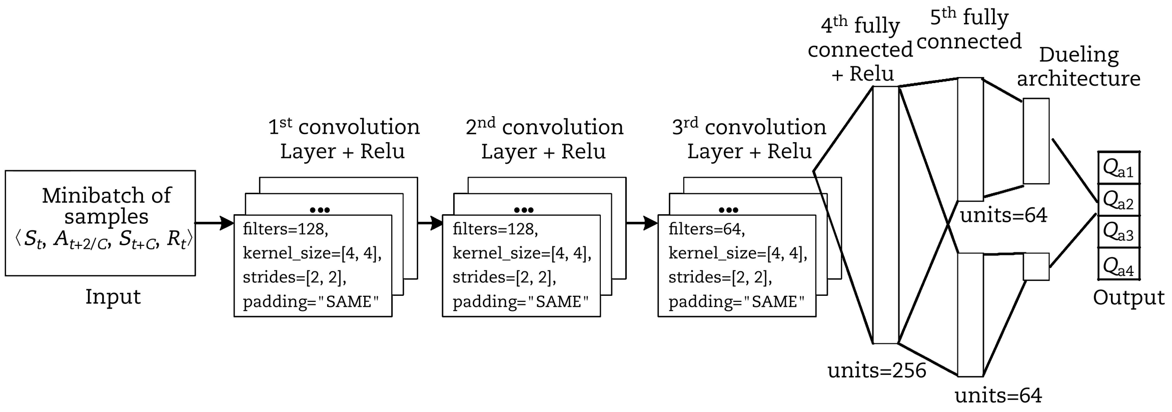
Minibatch of samples
$\langle S_t, A_{t+2/C}, S_{t+C}, R_t \rangle$
Input
1st convolution Layer + Relu
filters=128, kernel_size=[4, 4], strides=[2, 2], padding="SAME"
2nd convolution Layer + Relu
filters=128, kernel_size=[4, 4], strides=[2, 2], padding="SAME"
3rd convolution Layer + Relu
filters=64, kernel_size=[4, 4], strides=[2, 2], padding="SAME"
4th fully connected + Relu
units=256
5th fully connected
units=64
units=64
Dueling architecture
Q_{a1}
Q_{a2}
Q_{a3}
Q_{a4}
Output

Fig. 10. The architecture and hyper parameters of the 3DQN algorithm.

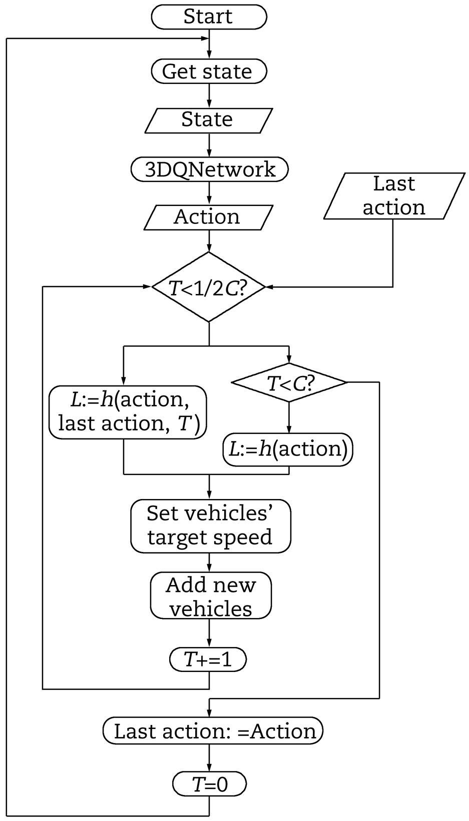

Fig. 11. The flowchart of the simulation3

for the next signal control cycle is generated by the current network in advance. The decision phase and the current phase determines whether to start the countdown timer. Subsequently, the decision phase will be started and held for half one cycle. Before the execution of the next signal cycle, the current network will be updated. The sample mini-batches is collected by the prioritized experience replay. And the target network is updated every few control cycles. The RMSProp Optimizer is used to update the parameters with the goal of minimizing the difference between the target Q-value and the predicted Q-value. intersection information.

The current network is copied to the target network to update the training goal every N step. The structure of our neural network is illustrated in Fig. 10. A convolution network is employed to approximate the policy function and duelling architecture, consisting of three convolution layers, two fully connected layers, as well as the hyper parameters, as shown in Fig. 10. The network outputs the Q-value of the four possible actions. The action with the maximum Q-value will be taken.

5. Experiment

In this section, we first compare the performance of the proposed AD-RLTSC algorithms with other traffic signal control methods in isolated intersections, including fixed time (FT), longest queue first (LQF), general reinforcement learning based TSC (RLTSC), Region-TSC and fixed time with bi-countdown timer (Bi-FT). Second, we apply the pre-trained model to a larger road network to explore the influence of the proportion of AD-RLTSC lights on large-scale mixed traffic flow.

5.1 Experiment setup

The experiment was conducted in a popular open-source traffic simulation software, SUMO [24].

Intersection: The whole simulation scenario is a 600 m × 600 m area. In simulations, an intersection with four ways and four phases was considered, as shown in Fig. 3. Each 300 m road consists of three lanes, controlling the left turn, the straight turn and the right turn, respectively. The right turn is always allowed, the grid size of the state matrix is 5 m, and the control cycle C is set to be 10 s.

Vehicle model: For HDVs, the Krauss car following model [25] acts as the default car following model and the CACC [26] model serves as the default model for the CAVs. The maximum speed is 20 m/s; the average acceleration and deceleration is 2.5 m/s^2; the vehicle length and minimum gap between vehicles are set to 5 m and 2 m, respectively.

Vehicle arrival rate: A sinusoidal function is used to simulate dynamic vehicle arrival rates. The control cycle $C = 10$ s; phase number $N = 4$; minimum time headway $h_s = 1.5$ s, and time loss of one phase $l = 1$ s. Fig. 4(a) suggests the vehicle arrival rate for each control cycle, ranging from 0.6 pch/s to 1 pch/s. In each episode, the whole simulation time is 15 h (54,000 s), and control cycle is 10 s, so the traffic light is controlled 5,400 times in one episode.

Table 2. Comparison of different traffic signal controllers on single intersection scenario.

Evaluation	FT	Bi-FT	LQF	RLTSC	Region-TSC	AD-RLTSC
Average vehicle waiting time (s)	8.26	8.75	7.2	7.04	7.02	7.0
Average vehicle speed (m/s)	9.66	9.55	9.57	10.00	10.54	9.89
Std. of vehicle speed	0.69	0.63	0.82	0.78	0.64	0.56
Dangerous acceleration (times)	201	0	219	248	223	0

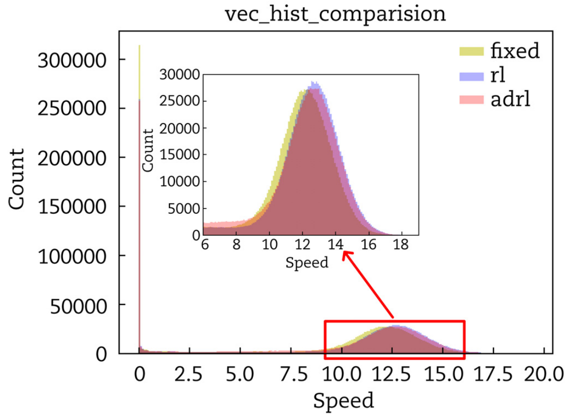

Fig. 12. Comparison on velocity distribution of all vehicles.

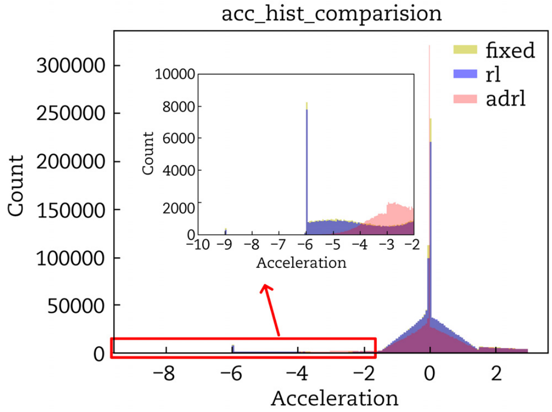

Fig. 13. Comparison on acceleration distribution of all vehicles.

Simulation process: Fig. 11 illustrates our overall simulation program flowchart for one control period. At the beginning of a decision cycle, the state matrix is extracted from the simulation software and fed into the input of the reinforcement learning model, namely 3DQ-Network. The action vector is a four-dimensional vector representing the probability of 4 phases. The phase with maximum probability is chosen but is not immediately activated. If the current phase is the same as the decision phase, the traffic signal state will not be changed. However, if the two phases differ, the countdown timer will be started to remind all vehicles that the current light signal will be in the reverse-phase mode when the timer ends. During the countdown, the target speed of all vehicles will be calculated based on the built-in car-following model using the induction method introduced in Section 3.2. Finally, new vehicles are added into the simulation environment and a one-step simulation is performed until the control period ends.

5.2 Single intersection

A comparison of different traffic signal controllers is presented in Table 2. The FT algorithm has the maximum queue length and leads to considerable dangerous behaviour. The Bi-FT algorithm solves the problem of dangerous acceleration by providing countdown information to vehicles. However, because of its fixed timing, it is a less-efficient approach. The performance of RLTSC and LQF is similar: they have better average vehicle speed compared to the former two methods, but they perform worse regarding both safety and stability. The Region-TSC further improves efficiency compared to the RLTSC, but still gives rise to dangerous driving phenomena. The proposed AD-RLTSC algorithm has the lowest vehicle waiting time. Finally, the Bi-countdown timer mechanism and the advance decision-making strike an effective balance between operational efficiency and dangerous acceleration.

Fig. 12 shows the velocity distribution for all vehicles. It can be seen that the vehicle speed is distributed mainly in two intervals, specifically, the interval of [10.0 m/s,15.0 m/s]. Any interval near 0 reflects the number of stops made by the vehicle. The FT has the maximum stop times and the RLTSC has the maximum vehicle speed. The AD-RLTSC has the minimum stop times and a relatively fast driving speed.

Fig. 13 shows the acceleration distribution. Most of the accelerations are in the interval of [2 m/s^2, 2.5 m/s^2]. A zero acceleration means the vehicle is driving with a constant speed. The AD-RLTSC has the largest zero acceleration distribution, which indicates that traffic flow is the most stable. The maximum deceleration is set to 6 m/s^2 and the maximum comfort deceleration is set to 3 m/s^2. If the deceleration is smaller than 3 m/s^2 and larger than 6 m/s^2, it means the human driver adopted an aggressive driving strategy. If the deceleration is smaller than 6 m/s^2, it indicates dangerous driving behaviour, especially in the case of a sudden red light. As can be seen, both FT and RLTSC methods often result in dangerous conditions. In contrast, the AD-RLTSC method provides comfortable acceleration most of the time.

To clearly clarify the behavior of vehicles near the intersection, Figs. 14 and 15 provide heat maps of vehicle speed and vehicle acceleration, respectively. The vertical coordinate is the position in the map and the middle position, set at 40 m, is the position of the intersection. The horizontal coordinate is simulation time, and there are 54,000 simulation steps in an episode. These two figures show vehicles' speed and acceleration on the straight lane running from north to south.

In Fig. 14, the red dot indicates that the vehicle speed is zero, which means the intersection is congested. The blue dot represents speed as the maximum speed, which means the intersection area is running smoothly. Fig. 14(d) the simula-

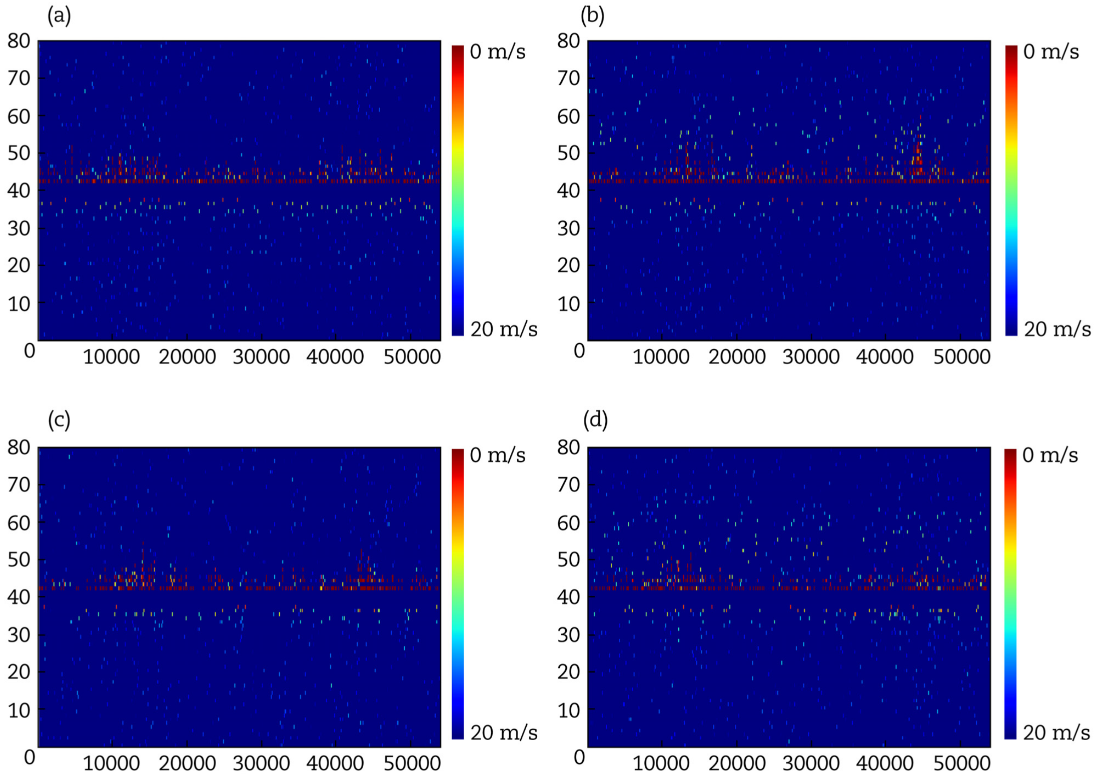

Fig. 14. Comparison on velocity heat map: (a) fixed time; (b) Bi-countdown; (c) reinforcement learning; (d) advance decision-making reinforcement learning.

tion results for the AD-RLTSC method. Compared to the other three methods, the congestion and queuing length near the intersection are significantly reduced, and the congestion is allocated to the upstream section.

In Fig. 15, the red dot indicates that the vehicle has a large acceleration, representing sudden braking. The blue dot represents a zero acceleration, that is, the vehicle is travelling at a comfortable constant speed. There is much less sudden braking in AD-RLTSC and Bi-FT due to the bi-countdown timer.

Overall, the proposed AD-RLTSC algorithm can improve intersection efficiency and help keep the traffic flow running smoothly.

5.3 Multiple intersections

We applied the pre-trained AD-RLTSC model into a 3 × 3 road network to evaluate its control performance in a large-scale traffic environment, as shown in Fig. 16. All signal lights operate independently in the multiple intersections scenario, and the traffic participants have both HDVs, CAVs, AD- RLTSC lights and FT lights.

An intelligent signal system penetration experiment was conducted to study the effect of the proportion of intelligent signal lights on overall traffic efficiency. The results are presented in Table 3. In this experiment, all the vehicles are HDVs. The traffic lights listed in the second row are controlled by the AD-RLTSC and the remaining lights are traditional fixed time signal lights. Result indicated that as the number of intelligent signal lights increased, the average speed of the vehicle increased, the average queue length at the intersection decreased, suggesting that traffic efficiency improved, and the standard deviation of vehicle speed and the queue length became smaller, indicating enhanced traffic stability.

As indicated in Table 3, the traffic efficiency is the highest when all the lights are AD-RLTSC. Table 4. shows the results of testing the effect of the ratio of CAVs on traffic efficiency. In this experiment, the AD-RLTSC algorithm controls all nine traffic lights. The results show that an increased ratio of CAVs leads to an increase in the vehicle moving speed and a decrease in intersection queue length. In short, it appears that CAVs also can enhance traffic stability.

Compared to the full HDVs and full FT controller scene, the full CAVs and full AD-RLTSCs scene improves the vehicle speed up to 15.9% and decreases the queue length up to 43%.

6. Conclusions and Discussion

In this paper, we proposed three new conceptualizations in the traffic signal control field.

The first is the Region-based RLTSC. Because the vehicle on the road will arrive at the intersection sometime in the future, we established the relationship between the perception range and the signal control period, and we defined a scalable state matrix, namely the trust region state. We considered the performance of the fixed time traffic light as the baseline to standardize the reward function of the reinforcement learning network. Based on the trust region state and the standardized reward function, we presented the Region-based RLTSC. However, rapid changes in signal lights still exacerbate the occurrence of dangerous behaviour. Second, to address this problem, which is common to most adaptive signal control methods, we presented the novel control approach, advance decision-making reinforcement learning traffic signal control. Compared with the general reinforcement learn-

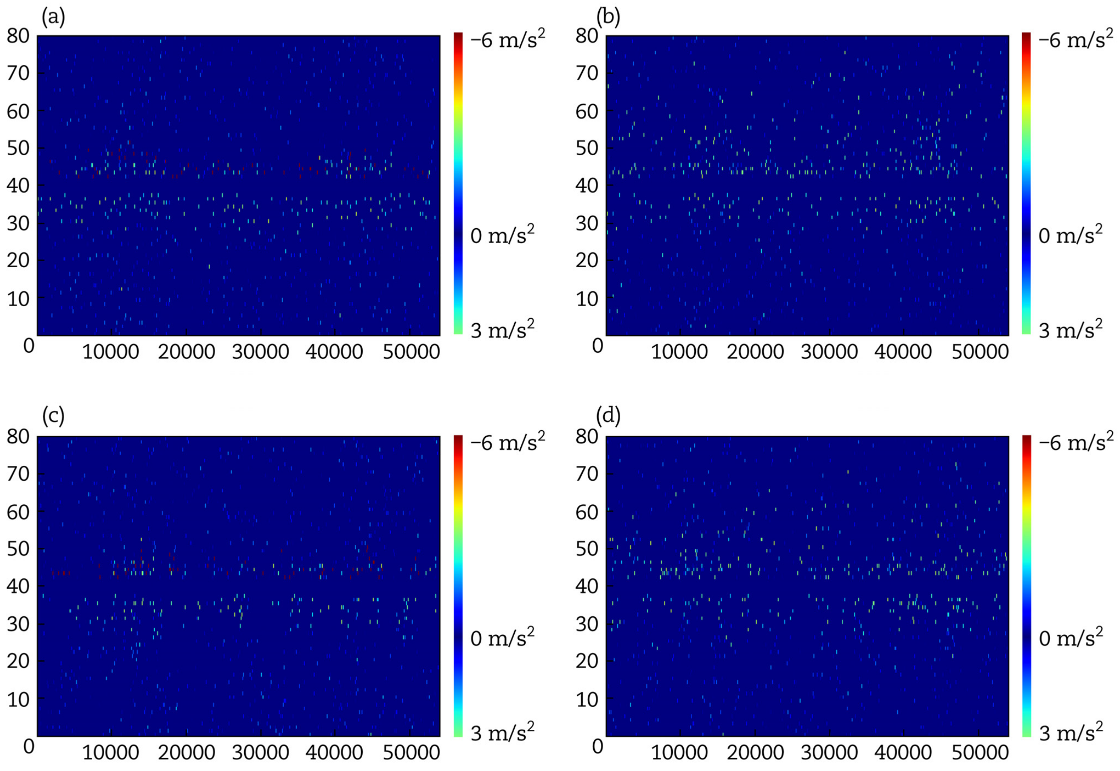

Fig. 15. Comparison on acceleration heat map: (a) fixed time; (b) Bi-countdown; (c) reinforcement learning; (d) advance decision-making reinforcement learning.

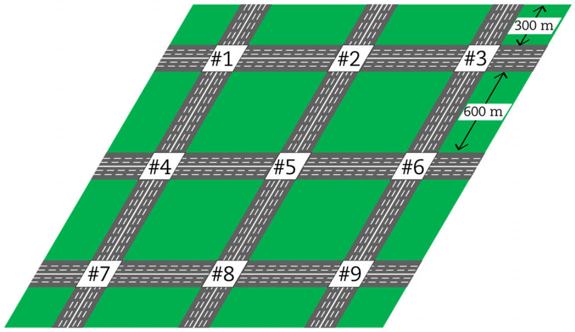

Fig. 16. Comparison on acceleration distribution of all vehicles.

ing algorithm, the AD-RLTSC separates the decision and control processes of the agent. Based on the theory of the trust region state, the information in a larger perception range can be used as the future state to make the future decision. The time difference between decision and control is used to broadcast future signal status to all vehicles to avoid dangerous driving behaviour. The bi-countdown timer is also proposed, which is a signal display mechanism that can be used in real-life scenarios. It consists of four states: green, red, green countdown and red countdown.

We conducted isolated intersection test and multiple intersections tests, and we compared the effects of different control methods on vehicle micro-behaviour and traffic macro-efficiency. The results showed that the proposed AD-RLTSC algorithm could simultaneously improve both traffic efficiency and traffic flow stability.

In the multi-intersection experiment, we evaluated the impact of both the intelligent signal penetration rate and the intelligent vehicle penetration rate on traffic efficiency. The results showed that traffic efficiency is improved as the number of smart subject increasing.

A methodological point worth noting is that in the present research the AD-RLTSC and the CAVs were always making the decision independently, although the CAVs could receive countdown information. In future work, we will combine the intelligent infrastructure and the intelligent vehicles together to improve traffic efficiency, safety and stability.

Table 3. The impact of ratio of AD-RLTSC signal lights on traffic efficiency.

Ratio of AD-RLTSC Light	**0/9**	**3/9**		**6/9**		**9/9**
The index of the AD-RLTSC	None	#2,#5,#8	#1,#5,#9	#1, #3, #4, #7, #8, #9	#2, #3, #4, #5, #6, #8	All
Average speed (m/s)	8.69	8.86	8.91	9.12	9.07	9.32
Std of vehicle speed	0.54	0.52	0.55	0.51	0.50	0.49
Average waiting length	47.47	41.57	42.34	36.26	35.67	30.56
Std of waiting length	17.47	15.74	15.41	14.55	13.90	12.15

Table 4. The impact of ratio of CAVs on traffic efficiency.

Ratio of CAVs	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Average speed (m/s)	9.33	9.38	9.42	9.50	9.56	9.62	9.70	9.78	9.87	9.97	10.07
Std of speed	0.497	0.504	0.502	0.504	0.497	0.494	0.491	0.483	0.474	0.464	0.460
Average waiting length	30.54	30.49	30.66	30.15	29.85	29.69	29.29	28.77	28.03	27.65	27.02
Std of waiting length	11.970	12.121	12.056	11.605	11.292	11.094	10.386	10.050	9.321	9.041	8.456

Acknowledgements

This work was supported by Science & Technology Research and Development Program of China Railway (Grant No. N2021G045), the Beijing Municipal Natural Science Foundation (Grant No. L191013) and the Joint Funds of the Natural Science Foundation of China (Grant No. U1934222).

Conflict of Interest

No potential conflict of interest was reported by the authors.

References

1. Zhu W-X, Zhang H. Analysis of mixed traffic flow with human-driving and autonomous cars based on car-following model. *Physica A: Stat Mech Applic* 2017; **496**:274–285.
2. Liu Y, Guo J, Taplin J, et al. Characteristic analysis of mixed traffic flow of regular and autonomous vehicles using cellular automata. *J Adv Transport* 2017; **2017**:8142074.
3. Chen J, Sun J. Platoon Separation Strategy Optimization Method based on Deep Cognition of a Driver's Behavior at Signalized Intersections[J]. *IEEE Access*, 2020; **8**(1): 17779–17791.
4. Qin Y, Wang H, Ran B. Impact of connected and automated vehicles on passenger comfort of traffic flow with vehicle-to-vehicle communications. *KSCE J Civil Engng* 2019; **23**: 821–832.
5. Chen J, Shangguan W, Cai B, et al. Communication Block Slot Optimization Method Based on Intelligent Vehicle Platoon Cognitive Ability Enhancement. *China Journal of Highway and Transport*, 2020; **32**(6):283–329.
6. Yau K-LA, Qadir J, Khoo HL, et al. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Comput Surv* 2017; **50**:34.
7. Gokulan BP, Srinivasan D. Distributed geometric fuzzy multiagent urban traffic signal control. *IEEE Trans Intell Transport Syst* 2010; **11**:714–27.
8. Prashanth L A, Bhatnagar S. Reinforcement learning with function approximation for traffic signal control[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2010; **12**(2): 412–421.
9. Tan T, Bao F, Deng Y, et al. Cooperative deep reinforcement learning for large-scale traffic grid signal control. *IEEE Trans Systems Man Cybern* 2019; 1–14. [Online]. Available: https://academic.microsoft.com/paper/2933570795.
10. Mohebifard R, Islam SBA, Hajbabaie A. Cooperative traffic signal and perimeter control in semi-connected urban-street networks. *Transport Res Part C: Emerg Technol* 2019; **104**: 408–427.
11. Aslani M, Seipel S, Wiering M. Continuous residual reinforcement learning for traffic signal control optimization[J]. *Canadian Journal of Civil Engineering*, 2018; **45**(8):690–702.
12. Abdulhai B, Pringle R, Karakoulas GJ. Reinforcement learning for true adaptive traffic signal control. *J Transport Engng* 2003; **129**:278–285.
13. Gregoire P-L, Desjardins C, Laumonier J, et al. Urban traffic control based on learning agents. in *2007 IEEE Intelligent Transportation Systems Conference*, 2007, pp. 916–921.
14. Liang X, Du X, Wang G, et al. Deep reinforcement learning for traffic light control in vehicular networks[J]. arXiv preprint arXiv:1803.11115, 2018.
15. Xu N, Zheng G, Xu K, et al. Targeted knowledge transfer for learning traffic signal plans, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Cham, Springer, 2019, 175–187.
16. Genders W, Razavi SN. Evaluating reinforcement learning state representations for adaptive traffic signal control. *Proc Comput Sci* 2018; **130**:26–33.
17. Tang J, Xiao Q. Comprehensive analysis of 5G coverage capability. *Radio Commun* 2019; **6**:28–32.
18. Kober J, Bagnell JA, Peters J. Reinforcement learning in robotics: a survey. *Int J Robot Res* 2013; **32**:1238–1274.
19. Akcelik R. Traffic signals: capacity and timing analysis. *Australian Road Research Board*, 1981.
20. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015; **518**: 529–533.
21. van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. in *AAAI '1 6 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2094–3100.
22. Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning, in *ICML '1 6 Proceedings of the 33rd International Conference on International Conference on Machine Learning – Volume 48*, 2016, pp. 1995–2003.
23. Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
24. Behrisch M, Bieker L, Erdmann J, et al. SUMOsimulation of urban mobility an overview. in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, 2011, pp. 55–60.
25. Krauss S, Wagner P, Gawron C. Metastable states in a microscopic model of traffic flow. *Phys Rev E* 1997; **55**: 5597–5602.
26. Milane´s V, Shladover SE. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transport Res Part C: Emerg Technol* 2014; **48**: 285–300.