ABSTRACT The conflict between limited road resources and rapid car ownership makes the traffic signal timing become a pivotal challenge. Emerging studies have been carried out on adaptive signal timing, but most of them still focus on the throughput of intersections, leaving safety and travel experience unconsidered. This paper proposes a time difference penalized traffic signal timing method by reinforcement learning technique to balance safety and throughput capacity in traffic control system. Firstly, a microcosmic state representation is proposed to integrate the dynamics of both traffic lights and road vehicles, including driver behaviors of lane changing, car-following, previous phase of traffic light and its duration. Secondly, an action space, including 8 signal phases, and a behavior-aware reward function are designed to resist the red-light overflow. Finally, a partial long short term memory (LSTM) network is trained to balance traffic efficiency and traveling experience. In the network training, a parallel sampling method is adopted to obtain experience from multiple environments to accelerate the training convergence in practical application. Experimental results show that the proposed method improves the intersection efficiency up to 14.28% compared to the fixed signal timing and 5.26% compared to DQN while getting rid of red-light overflow time.

INDEX TERMS Traffic signal timing, deep reinforcement learning, DQN, LSTM, driving behavior, driving safety.

I. INTRODUCTION Today, the global car ownership in the world has exceeded one billion, which led to a series of serious problems, such as traffic congestion, environmental pollution, energy-wasting etc. [1], [2]. Intersection, as a key node in the urban traffic network, plays a pivotal role in traffic manage and its optimization. Growing attention in academic and industrial fields has been paid to the traffic light control at intersection [3]. However, the limitation of road resources and the complexity of dynamical traffic flow make it hard to optimize signal timing to improve the capacity at intersection while keeping safety [4]. Intelligent traffic signal control is a measure to deal with space conflicts, which not only reduces environmental pollution and energy waste [5], but also improves driving efficiency. How to make full use of the limited resources at intersections is becoming a severe challenge in intelligent traffic management [6].

Signal timing is a collaborative optimization of spatio-temporal resources. The state-of-art methods are generally carried out under a predetermined scheme, with which the intersection capacity is usually hard to be fully exploited [7]. Few of them focused on the conflict-avoidance while improving the capacity of intersections [8], [9].

The blooming development on artificial intelligence has brought growing efforts in the field of adaptive control of...
signals [10]. Through reinforcement learning (RL), an agent interacts with the environment and explores reasonable action. RL has been employed as a promising solution to optimize traffic signal timing and its control without relying on the manual phase planning. Various signal timing methods have been proposed to explore more efficient signal timing. But most of the existing works focus on one goal to maximize the throughput of intersections [11]. Reported from the U.S. Department of Transportation, overflow length of red light is a crucial factor to the safety at intersections [12]. If the traffic flow is unbalanced at an intersection, the busy lane will tend to get unlimited priority to maximize reward and the free lane will encounter an unbearable waiting time, which may lead to drivers’ illegal driving through red lights. On the contrary, a short signal cycle is a barrier for improving the efficiency of intersections [13]. Therefore, it is a difficult trade-off between safety and capacity in an adaptive signal control system. Worse still, existing methods have not considered lane changing, which brings a negative effect on intersection efficiency [14].

Aiming at these issues, we propose a novel intelligent signal timing method named Rep-DRQN to process signal control in this work. After a short view of the system framework, a state representation with lane-changing sensibility and an action space of full flexibility are proposed to catch traffic state at intersections and take the most reasonable actions. Then, a novel reward function is designed with RELU(Rectified Linear Unit) to deal with the intolerable waiting time for green light. Finally, to get efficient processing in complex traffic scenarios, we adopt a parallel learning scheme to accelerate the training process.

This paper is organized as follows: the related works on the signal controlling are reviewed in Section II. Then, we propose the system modeling and methodology for intelligent signal timing through deep reinforcement learning in parallel environments in Section III. Finally, we introduce the experimental results and discussion in Section IV and conclude this work in Section V.

II. RELATED WORKS

Various RL methods have been proposed for traffic signal control. In early study, due to limited resources, researchers had to simplify the situation of intersections as much as possible [15]. As example, Thorpe used four elements in the state space, including vehicle count, fixed-distance, variable-distance and duration [16], and then the performance of two SARSA schemes are analyzed [17]. Abdulhai et al. defined the state as the queue lengths on the four approaches and the elapsed phase time, and reward function as the total delay of the four approaches [18]; then Q-learning is used to select whether to remain or change the phase. These methods simplify the traffic scene to make real-time calculations possible. However, the simplified methods lose some important features of the traffic state.

With the rapid development of computing devices, several methods emerged in RL. Konda and Tsitsiklis [19] proposed Acto-Critic method, in which the actor is used to choose the action, while the critic is used to evaluate the chosen. Under this framework, Aslani et al. [20] designed an adaptive traffic signal controller (A-CAT controller) and applied it to the city center of Tehran. Li et al. [21] applied the stacked auto-encoders (SAE) to estimate the Q-function with a deep neural network. The action space includes two signal phases, and the average delay can be reduced about 14%. Gao et al. [22] further used a convolution neural network to learn the optimal policy, and the algorithm reduces vehicle delay by up to 47% and 86% when compared to another two popular traffic signal control algorithms, longest queue first algorithm and fixed time control algorithm, respectively. These methods learn from traffic scenes and make decisions automatically to improve traffic efficiency at intersections.

To guarantee training convergence, researchers pay their attention to maximize profit from refined modeling and intelligent controlling. Xu et al. [23] used a deep recurrent Q-network(DRQN) to fit the state and Q-value. The state was defined as a five dimension vector, including the number of vehicles and their average speed in a current intersection, number of vehicles and average speed in the neighbor intersection, and the signal state. The action included phase change and phase invariance. And the reward was the average delays. Casas [24] employed deterministic policy gradient (DDPG) to receive the information of all detectors and generate the timing of all traffic light phases. Genders and Razavi [25] used the asynchronous n-step Q-learning algorithm to reduce the average vehicle total delay by 40% without affecting throughput. The state-of-art adaptive traffic signal control are summarized in Table 1.

These methods above have achieved considerable improvement of increasing road throughput, but rare research focuses on the traffic compliance at the intersection and drivers’ travel experience [26], which is crucial for modern traffic management. In this work, we propose a reward-penalty traffic signal timing with deep reinforcement learning.
learning. The state space is modeled with multiple dimension, including lane changing behavior and red light waiting time. A full action space includes all possible choices. The reward function is the response to excessive waiting in red light, we construct a partial memorial neural network to balance traffic efficiency and travel safety.

III. SYSTEM MODELING AND METHODOLOGY

A. PROBLEM DEFINITION

Definition 1 (Signal-phase Vector (SV)): A typical four-way intersection in right-hand traffic is studied in this paper, and it’s assumed that each way has three entry lanes, in which the right one for turning right, the middle one for going straight and the left one for turning left. All 12 entry lanes in the intersection are tagged as from L1 to L12, whose order is shown in Fig. 3, and the signal status of all entry lanes are denoted as vector <L1, L2, ..., L12>, which is named as Signal-phase Vector (SV).

The signal status is represented as color, including Red, Green and Yellow, so SV is a color series. For example, <GGGGRRRGGRR> means that the signals in lanes from L1 to L4, lane L7 and L10 are green and the others are red. It is noteworthy that four turn-right lanes, including L1, L4, L7 and L10, are assumed as permanent green because there is no conflict between them and the others.

Definition 2 (Action Vector (AV)): The signal action is to set signal color in all lanes, so, like SV, the action is represented as a vector of destination color series. This vector is named as Action Vector (AV). It is noteworthy that the color change in a lane automatically insert a fix-length Yellow phase between the former and the latter, so there is no Yellow phase in all action vectors.

Definition 3 (Road-segment State Vector (RSV)): Each lane in the intersection is divided into equal segments by a length of 2 meters, and each segment is described as a three-dimensional state vector. This state vector is named as Road-segment State Vector (RSV), in which <0, 0, 0> indicates that there is no vehicle on the road segment; <1, 0, 0> indicates that there is a vehicle with left steering-lamp on in the segment; <0, 1, 0> indicates that there is a vehicle traveling straight, and <0, 0, 1> indicates that there is a vehicle with right steering-lamp on.

Definition 4 (Bearable Red-light Duration (BRD)): In an unbalanced traffic, keeping green light in some lanes might increase intersection’s throughput, but it may lead to an endless red light in some other lanes. To avoid the unbearable red light, the Bearable Red-light Duration (BRD) is set as a custom parameter to describe travelers’ different tolerance degree on red light duration, and we set BRD as 300 seconds in this work.

Definition 5 (Excessive-duration Marking Vector (EMV)): To make signal timing system sensitive to long red-light durations at each entry lane, 12 lanes are marked whether it appears a longer red-light duration than BRD, and these Boolean markings form a 12-dimensional vector, named as Excessive-duration Marking Vector (EMV).

B. SYSTEM FRAMEWORK

An ideal control system of urban traffic signal should own the capability to respond to dynamical traffic flow and make online optimization in time [27]. In this work, RL is employed to learn from traffic flow at intersection and then optimize signal timing online. The RL framework in this work, as shown in Fig. 1, could be divided into two parts including system agent and its environment. The environment is on behalf of the real traffic scene simulated by SUMO, which is the most popular traffic simulation platform developed by Krajzewicz et al. [28]. The agent is the signal control which owns the ability to observe environments and to take actions on their own.

The basic idea of RL is that the agent takes an action $a_n$ according to the current strategy on state $s_n$ of the environment. After action executed, the environment transfers to the next state $s_{n+1}$. At the same time, the reward $r_n$ of this action is fed back to the agent, from which the agent could evaluate the effect of the last action and improve the strategy while exploring the environments. The agent constantly adjusts its strategy until the optimal strategy is found eventually.

In a RL system, the state space, action space and reward function should be defined elaborately. In intelligent traffic signal control, the state space should be full of information reflecting traffic flow patterns deciding different signal...
timing, the action space should include all kind of signal phase without conflict, the reward function should evaluate each action taken by agent with a reasonable value. With these three key models being defined, a deep neural network with suitable structure is designed to find the relationship between states and actions, i.e., \( f(\text{state}) = \text{Q-value} \).

### C. SYSTEM MODELING FOR REP-DRQN

#### 1) STATE REPRESENTATION

In an intelligent signal system, the control agent could choose reasonable actions based on observed states, only if the agent could be sensitive to environmental changes in state representation. That’s why the state representation needs to include all key potential features of the traffic flow at intersection. These key features include vehicles’ state, expressed as the spatial distribution of vehicles. Some modeling methods of vehicle distribution have been proposed, e.g., a spatial matrix. Liang \textit{et al.} proposed a modeling method to generate a matrix to represent vehicle distribution at the intersection, which contains vehicle position and velocity information. Then a convolutional neural network is used to obtain the traffic flow features \[29\]. However, this modeling transforms all the space covering intersection into grids to form a matrix, no matter it’s in the lane or off the lane. As consequence, this modeling generates lots of redundant data, which decreases the training efficiency and convergence speed.

In this work, we pay attention not only to the vehicles’ position distribution but also to the drivers’ lane changing behavior. We construct a matrix with the vehicle location labeled with lanes and the distance to intersection. Besides, the lane changing behavior is also been taken into consideration. We pay attention to the vehicle’s steering lamp and include it into the matrix which enables the intelligent agent to understand drivers’ behavior. Each lane is divided into several segments according to the distance and expressed as RSV. When there is a vehicle at a certain distance in the lane, the corresponding element in RSV is marked as 1, otherwise 0. However, when a vehicle appeared on the road network for the first time, the occupied lane segment changes. In this case, the status of the vehicle’s steering lamp is fed to the neural network, to indicate the future direction of vehicles. In details, we consider RSV as the frist state \( S^1_t \). The specific representation is also shown in Fig. 2.

### TABLE 2. Action space of signal phase.

| Phase code | Direction of traffic flow | Color series of signal light (L1~L12) | Graph of traffic flow |
|------------|---------------------------|----------------------------------------|----------------------|
| \( a_1 \)  | N-SL                      | GGGRGRGRGRGR                           |                      |
| \( a_2 \)  | E-SL                      | GRGRGRGRGRGRGR                         |                      |
| \( a_3 \)  | S-SL                      | GRRGRRGRRRRR                           |                      |
| \( a_4 \)  | W-SL                      | GRGRGRGRGRG                            |                      |
| \( a_5 \)  | SN-L                      | GRRGRGRRGG                            |                      |
| \( a_6 \)  | SN-S                      | GRRGRRGRGRR                           |                      |
| \( a_7 \)  | EW-L                      | GRRGRRGRGRG                           |                      |
| \( a_8 \)  | EW-S                      | GRRGGRGRGRG                           |                      |

Except for making full use of the resources of the intersection, the psychology of driver should also be considered to improve traffic safety. As reported from the Federal Highway Administration in the U.S. Department of Transportation, if a driver waits too long at intersection, his patience might be exhausted and then violate traffic regulations \[12\]. Therefore, we proposed a method to make the neural network sensible to drivers’ waiting time and then tune a suitable waiting time to balance traffic efficiency and safety. We construct a vector \( S^{(2)}_t \) with EMV, which is the red-light duration beyond BRD in each lane. If the current signal time is less than BRD, EMV will be defined as zero. The formula of EMV is shown in (1):

\[
\text{EMV}_i = \max(L_i - \text{BRD}, 0)
\]

where, EMV\(_i\) is the excessive-duration of lane \( i \), \( L_i \) is the current duration of the red light of lane \( i \), and BRD is a bearable duration waiting for green light, which could be adjusted according to practical condition. These states are transmitted to the neural network, and the network would make a response by updating the weight and bias.

Additionally, considering that the current signal is also an important factor on the decision, we add the current signal phase as the third part of status representation in the system, denoted as \( S^{(3)}_t \). \( S^{(3)}_t \) is a vector representation encoded with one-hot encoding \[30\] for current signal phase, which is encoded from 8 non-conflict signal phases shown as actions in Table 2. It’s worth noting that the yellow light is only an isolation phase between different phases, which does not play a role in relieving traffic flow so the yellow light is skipped while taking the previous signal phase.
2) ACTION SPACE
In order to improve the efficiency of intersection, the intelligent agent needs to choose a proper action from action space, so the flexibility of action space would make a considerable effect on the feasibility of the agent. In this work, we manage to make a most flexible action space for the intelligent control scheme of signal lights. First of all, we assume that each lane has its own direction and label, which is shown in Fig. 3.

We went through and checked all possible signal phases for each lane in the intersection. With two lanes except right-turning lanes being green light at the same time, there are 8 non-conflicting signal phase pairs, which are represented as color series shown in Table 2 and employed as 8 actions in DQN. The collection of these 8 actions forms an action space, and as shown in Table 2, these 8 actions are marked as a1 ~ a8. It is noteworthy that we take no account of the right-turning lanes because they have no conflict with others and are always accessible.

In the actions table above, the direction of traffic flow is defined as two parts, i.e. traffic entrance direction and vehicle allowed directions. The color series of signal light corresponds to the lane label L1~L12 (as defined in Fig. 3) in turn. For example, for action a2, the direction of traffic flow is E-SL, which means that the vehicles entering from the east(E) can go straight(S) and turn left(L). And the corresponding color series of signal light is <GRRGGGGRRGR>.

Through these signal phases combinations, 8 actions can be selected at each timestep according to the current state. Besides, an 8-dimensional vector is built with one-hot encoding to represent the current signal phase, which is the third part of input data of status, symbolized as $S^j_i$ (5). For example, the phase status $S^j_i$ of W-SL could be represented by a vector <0, 0, 0, 1, 0, 0, 0, 0 >.

When an action is the same as the previous one, it means prolonging the length of current phase. We set the timestep as a small value for adjusting phase with most flexibility, which is one second in this work. Note that when signal phase changes, the environment will insert a yellow light before a new signal light.

3) REWARD WITH A PENALTY FOR RED-LIGHT OVERFLOW TIME
The reward function is employed to evaluate the selected action of each timestep. A reward function lacking of full consideration usually leads to a bad result in training. In some researches, the reward function were designed to deal with the issue of traffic efficiency. Liang et al. set the reward as the cumulative waiting time between two neighboring intersections [29], Mousavi et al. set the reward as the total delay of the vehicle in two adjacent movements [31]. In this work, we use the throughput of the intersection as a reward function. However, the observation duration is too short, just one second in this work, leading to the issue that the throughput of the intersection in one timestep is too small to be calculated. So, we consider the throughput of the intersection as follows.

$$r_t = \sum_{i=t-\Delta t}^{t} T_i$$  (2)

where $T_i$ represents the throughput of intersection at timestep $i$. Since the traffic light does not affect the right-turning vehicle, the reward will exclude the number of right-turning vehicles. The reward function is represented as the throughput of the vehicle at the intersection within the sliding time window $\Delta t$. Notably, during phase switching, there is still a possibility of throughput increasing. So $\Delta t$ should include the duration of yellow light.

In addition, as discussed above, avoiding excessive long Red-light duration is critical to balance the traffic efficiency and safety at the intersection, so we design a corresponding penalty item in the reward function, which will accumulate red-light duration exceeding BRD from each lane and calculate its square root to avoid a sharp switch. The final reward function is shown in (3):

$$r_t = \sum_{i=t-\Delta t}^{t} T_i - \frac{1}{2} \sum_{j} \max(L_{jt} - BRD, 0)$$  (3)

where $L_{jt}$ is the red signal duration of lane $j$ at time $t$. The square root is to avoid a sharp cutting of time and keep a more smooth changing for safety.

D. METHODOLOGY
1) REWARD-PENALTY DQN DESIGN
The optimal signal phase can be solved through Q-learning, with the state space, action space and reward function defined above. The Q-value is updated by (4).

$$Q(s_t, a_t) = \max \left[ r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a \right]$$  (4)

where $Q(s_t, a_t)$ is the Q-value of action $a_t$ at timestep $t$, $r_t$ is the reward value of the action $a_t$, and $\gamma$ is the greedy factor, generally between 0.8 and 1. With a larger $\gamma$, the agent would pay more attention to the subsequent reward, while, with a smaller $\gamma$, the agent would pay more attention to the current reward.
If the state space of the environment is tremendous, it will be a nightmare to deal with original Q-learning, for that it’s hard to converge. Deep Q-Learning (DQN), a combination of deep learning and RL, was proposed to fit the Q-values, using the loss function to adjust the weight and bias of the neural network. Now, DRL has been widely used in various fields, such as the game of Go, self-driving, and signal timing. In this work, we employ DQN to train intelligent agent from interaction with environments for its continuous state, whose loss function is shown as in (5):

$$L_i(\theta_i) = \mathbb{E}_{S_t, a, r, S_{t+1} \sim U(D)}[(r + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}; \theta_i^-) - Q(S_t, a_t; \theta_i))]$$

where $\theta_i$ is the parameter of the neural network at the $i$-th iteration, $U(D)$ is the sample set from intersection with multiple environments, and $\theta_i^-$ is the target parameter of the neural network. In this way, the parameters of the neural network will be updated iteratively in the intersection between agent and environment, which is shown in Fig. 4 below. Finally, the Q-value obtained by the neural network will be approximate to the real Q-value.

We add a partial long short term memory network (LSTM), since traditional recurrent neural networks would raise the problem of gradient disappearance or explosion. Therefore, we characterize the three parts of input data according to their features and endow the road situation state in the first part with memory ability because of the heavy correlation between successive road situations. The neural network proposed in this work is shown in Fig. 5, in which the state $S_t^{(1)}$ is propagated to LSTM network to solve the problem of gradient disappearance or explosion.

2) MODEL TRAINING AND LEARNING
In DQN’s environment, the experience data gained from multiple environments are collected and stored in a replay memory $D$ for agents’ sampling. The replay memory is an allocated memory storing experience data in DQN and allowing us to reuse these data later; the experience data include the last states, last action, current reward for the last action, and the current state ($S_t, a, r, S_{t+1}$) generated from the last action. After the replay memory has reached certain capacity, the agent will randomly extract a part of experiences to break the correlation between the experiences and calculate the temporal difference error (TD-error). Finally, the Q-value from the neural network will be approximate to the target Q-value.

However, only one experience item ($S_t, a, r, S_{t+1}$) could be generated in each iteration. But lots of experiences are required in the DQN algorithm, making its time complexity tremendous. In 2016, Google Deep Mind proposed an asynchronous method [32] to generate experiences ($S_t, a, r, S_{t+1}$) simultaneously with multiple environments. In this work, we similarly generate experiences by interacting with multiple environments in parallel.

In RL, enhancing data independence is an important way to improve data productivity and training efficiency [33]. Unfortunately, in urban traffic signal control, although experiences could usually be generated by multiple environments at the same time, there are various uncertainties in the environment, which may still contain some relevant experience. Therefore, we propose an asynchronous sample method to achieve a quicker convergence and a better guarantee for irrelevance between experiences. After multiple environments have gained their own experience, all the experiences will be gathered together and put into replay memory $D$ for agents’ sampling as shown in Algorithm 1. It’s worth noting that we do not set the termination signal in Algorithm 1, because there is usually no termination feature in a traffic signal control system.

As shown in Algorithm 1, the neural network is trained from inputting experience data saved in the memory $D$. In this method a probability factor $\epsilon$ is provided to avoid falling into a local optimum with partial random. We randomly generate a value from 0 to 1. When this value is less than the probability factor $\epsilon$, we randomly choose an action, otherwise we select the action corresponding to the maximum Q-value. In Algorithm 2, the state $S_t$ is sampled from the parallel generations of multiple simulation environments, the action $a_t$ is selected by the agent according to the state $S_t$, and the environment gives feedback a reward $r_t$ and then generate the next state $S_{t+1}$. All of these data will be saved as a vector ($S_t, a_t, r_t, S_{t+1}$) in the replay memory.
Algorithm 1 Training of Rep-DRQN

| Input | replay memory $D$ with capacity $N$ |
|-------|----------------------------------|
| Output | parameters $\theta$ of network |

repeat
1. if the length of replay memory $> maximum capacity$ then
2. sample random minibatch $((S_t, a_t, r_t, S_{t+1})$ from $D$
3. Set $y_t = r_t + \gamma \times \max \hat{Q}(S_{t+1}, \ a'; \ \theta)$
4. Preform gradient descent step on $(y_t - \hat{Q}(S_t, a_t; \ \theta))^2$ to the network parameters $\theta$
5. remove the former experiences of replay memory $D$
end
8. until Training finished;

Algorithm 2 Experience Data Generation With Rep-DRQN

| Input | replay memory $D$ with capacity $N$; neural network |
|-------|----------------------------------|
| Output | experiences of replay memory |

1. Create multiple simulation environments
2. For each simulation simultaneously repeat
3. Initialize the simulation of traffic
4. for timestep = 1,$M$ do
5. Select a random color $a_t$ with probability $\epsilon$
6. otherwise set $a_t = \text{arg max } Q(S_t, a_t; \ \theta)$
7. if $a_t \neq a_{t-1}$ then
8. Perform the yellow signal in 3 seconds
9. end
10. Perform the chosen color $a_t$ in a second
11. Obtain the reward $r_t$ and next state $s_{t+1}$
12. Store transition $(S_t, a_t, r_t, S_{t+1})$ in $D$
end
14. until Training finished;

After enough experience generated, the training module as Algorithm 1 will start to train the neural network, while the Algorithm 2 work simultaneously to generate experiences consecutively.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

After system modeling and algorithm designing introduced, system implementation and its experimental results are shown in detail in this section.

A. EXPERIMENTAL ENVIRONMENT

In this work, the microscopic traffic simulation software SUMO is employed to construct the experimental environment of traffic, and the Tensorflow is used to build the neural network for deep learning framework. The operational environment for simulation experiments is a 64-bit computer with 16GB memory equipped with a GPU processor for training the neural network, which is NVIDIA Geforce GTX 1080Ti with 11GB memory.

B. SYSTEM EXPERIMENTS

1) STATE GENERATING

As introduced in Section III above, the state data for inputting to the neural network consist of three parts, including the state of road and vehicles, driving behavior, and colors of the traffic light in the last phase.

Assuming that each lane of every leg at the intersection indicates a specific turning, we assign the direction of lanes as left turn, straight ahead and right turn in order as shown in Fig. 3, and we then divide each lane into 50 segments equally by the length of 2 meters to code the road states and position vehicles. Owe to the vehicle positioning with this gridding, the state $S^{(1)}_t$ could save the vehicle information for the last 20 seconds. In each position of the lane, there are four different states of vehicle, including no vehicle, vehicles with the steering lamp off, vehicles with the left-steering lamp on and vehicles with the right-steering lamp on. So every road segment could be coded as a RSV. The vehicle states $S^{(2)}_t$ could be limited into a finite state space. The state space is expressed as a 85 bits dimensional vector (20, 12, 50, 3), which means the traffic status in the last 20 seconds, 50 segments in all the 12 lanes, and 3 kinds of states in each road segment.

As mentioned above, the second and third part of state $S^{(2)}_t$ and $S^{(3)}_t$ are for the current red light duration and colors of the last phase. Among them, the $S^{(2)}_t$ is the red signal duration of 12 lanes corresponding to the traffic lights. Since the right turn signal are always a green, the value of the corresponding states $S^{(2)}_t$ of these lanes is always 0. In the simulation, we set the total traffic simulation time as 6000 seconds to get a moderate balance between experiment duration and its effect. It’s worth noting that within the first 20 seconds there is not enough state data as input, for which we fill the lacking state in the first 20 seconds with zeros and ignore saving it into the replay memory. In the third part, we generate the state by saving the last action for that $S^{(3)}_t$ is equal to $a_{t-1}$. Then, we employ 12 simulation environments to generate transitions simultaneously.

2) DEEP NEURAL NETWORK AND RL

Due to the fact that only the $S^{(1)}_t$ has strong spatial-temporal relation, we design a LSTM network with 2,560 cells. The other two parts are connected to the full connected layer directly. The deep network has two fully connected layers with 10,240 and 2,560 neurons respectively. The output is Q-values of 8 colors. We randomly extracted 64 transitions from the replay memory as input to the neural network, and then use algorithm AdamOptimizer to optimize the neural network with a learning rate of $10^{-6}$. We set the discount factor $\gamma$ as 0.99 and randomly select an action with a probability of linear decrement $\epsilon$ from 0.01 to 0.0007. Meanwhile, we set the initial $\epsilon$ as a low value, so that the color alternating is not reasonable before training. Then, we calculate the reward by counting the vehicles going straight and taking left-turn. Considering that the vehicle could change lanes, we exclude these vehicles changing lane, and calculate the throughput as...
shown in (6): 

$$T_i = \{veh_i | veh_i \in SL_t \& veh_i \notin RSL_{t+1}\}$$ (6)

where \(T_i\) means the number of vehicles passing through the intersection excluding right-turn at time \(t\); \(SL_t\) represents the total vehicles on the straight and left-turn lanes of entering the intersection at time \(t\); \(RSL_{t+1}\) represents the vehicles on the straight, left and right turn lanes of entering the intersection at time \(t + 1\). Finally, we calculate the vehicle throughput during the latest 20 seconds as a part of the reward function. In order to prevent a long-time red light, a negative reward is given if the red signal duration exceeds BRD. It’s worth noting that it is different between setting this penalty function for red-light overflow and the fixed maximal phase length, because the penalty function helps to change phase smoothly and balances the traffic safety and the capacity of intersection [12].

$$r_t = \sum_{i=t-19}^{t} |T_i| - \sqrt{\sum_{j} \max(Ljt - 300, 0)}$$ (7)

where \(Ljt\) represents the red signal duration of the lane \(j\) signal at time \(t\).

C. RESULTS AND DISCUSSION

Focusing on the traffic efficiency and its safety in the intersection, we mainly capture the statistical information of vehicle throughput, waiting number, and travel time, to evaluate the system performance. All the data demonstration and its analysis are shown in the following section.

1) TRAFFIC EFFICIENCY

As two typical methods, DQN and the fixed signal timing (FST) are employed as baselines in this work. FST is a traditional method for signal timing, which is also widely used now, and DQN is the main method employed for adaptive signal timing recently. As a typical setting, each phase duration of FST is set as 30 seconds; its action is limited to 4 phases, which are a1, a2, a3 and a4 shown in Table 2, and its phase changing will insert a yellow light for 3 seconds.

After every episode, we counted the number of vehicles going straight and turning left through the intersection, and then compare it to the baseline of DQN and FST. After 250 episode, the throughput and waiting number of vehicles are shown in Fig. 6 and Fig. 7 respectively.

As shown in Fig. 6, the throughput with FST is fluctuating at 3,500. With DQN, the throughput is up to 3800 after 100 epochs, but with Rep-DRQN, the throughput of vehicles are mounting up to 4,000 after 100 epochs, and Rep-DRQN tends to be convergence after 150 epochs.

The comparison of waiting queues is shown in Fig. 7, whose waiting number is the average number of vehicles waiting for green light in lanes. As shown in the figure, FST’s waiting number is about 65; DQN’s waiting number decreases to 58 after 100 epochs, and Rep-DRQN’s decreases to 40 after 150 epochs.

Further, the halting time and travel time are compared among these methods. As shown in Fig. 8, FST’s halting time is about 75 seconds, and its travel time is about 105 seconds; DQN’s halting time decreases to 60 after 100 epochs, and its travel time decreases to 80. In contrast, Rep-DRQN’s halting time decreases to 49 seconds after 150 epochs, and its travel time decreases to 60 seconds. It could be seen that with the proposed Rep-DRQN method vehicles could pass through the intersection about 45 seconds less on average than FST and 20 seconds less than DQN, which would significantly improve travel experience and contributes energy saving.

2) TRAVEL EXPERIENCE AND SAFETY

Besides aiming at the efficiency of intersection, safety is another challenge for the controlling strategy of intelligent signal timing. Travel experience pays more attention to individual behavior, which is often affected by the driving environment. So, travel experience is one of the ways to reflect safety at intersections. According to the driver’s travel experience, the excessive length of red light duration is the major factor to galvanize drivers to break traffic rules with exhausting patience. Therefore, we analyze the pattern of red light duration for the evaluation of safety. After training of 250 episodes, we conduct a system test on the generated neural network for its red light duration in one episode. In the experiment, 3,961 vehicles are going straight ahead or turning left, whose duration pattern of red lights in all lanes is shown in Fig. 9.

From the distribution of red light duration, it could be known that little phase of red light is beyond 300s. As could be known above, In addition, there will be a certain penalty in the reward value of the agent if the red light duration is
as shown above and the negative Q-value will accelerate signal’s switching to interrupt a red-light overflow.

In this result, the first Q-value 177.25 is relatively larger than the others. It means that Rep-DRQN tends to take the action \( a_1 \) to change the red signal in L2 and stop its overflowed red time. Meanwhile, since the action is selected every second for precise controlling, a reasonable signal timing should be kept long enough. So, we further analyze the transition characteristic between different phases with transition matrix, which is a common expression for analyzing state changes. As shown in Table 3, we generate a transition matrix of action selections between two adjacent phases. First, we use these 8 kinds of actions as input by the third part of state data state \( S_t^{(3)} \), and then observe the output as Q-values, which is used as the value of the transition matrix and recorded in Table 3.

From the transition matrix, it is shown that in general every action possesses the biggest Q-value for a transition to itself, which makes it possible to last duration for a suitable length. Additionally, it is a remarkable fact that there is the most possibility to transfer from action \( a_6 \) to action \( a_1 \). Our further analysis for this special case shows that there is an overlapping action in the action \( a_1 \) and \( a_6 \) as shown in Table 2, and also shows that the action transition does not fully depend on the previous phase.

### V. CONCLUSION

Making a balance between traffic efficiency and safety at intersections is still a pivotal challenge in intelligent traffic management and smart city building. In this work, we propose a signal timing method named Rep-DRQN, in which we refine state modeling with multiple dimension, including driving behavior of changing lanes and drivers’ waiting time. We construct a full action space with all possible choices and define reward function with response to excessive waiting in red time. Finally, we design a partial LSTM neural network to balance traffic efficiency and travel safety.

The experimental results show that the traffic capacity at intersection with Rep-DRQN is improved by 14.28% compared with FST and 5.26% compared with DQN. And the maximal phase length in red time is dynamically maintained, which brings drivers better travel experience and safety, and reduces both energy consumption and environmental pollution at the same time. It benefits for practical application with
intelligent signal timing by transferring the onediced pursuing efficiency to practical requirements.

ACKNOWLEDGMENT
The authors would like to thank the Fujian Provincial Big Data Institute for Intelligent Transportation, the Fujian Key Laboratory of Automotive Electronics and Electric Drive, and the Fujian Provincial Department of Transportation for supporting the experiments.

REFERENCES
[1] L. Qi, M. Zhou, and W. Luan, “A two-level traffic light control strategy for preventing incident-based urban traffic congestion,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 1, pp. 13–24, Jan. 2018.
[2] L. Qi, M. Zhou, and W. Luan, “A dynamic road incident information delivery strategy to reduce urban traffic congestion,” IEEE/CAA J. Automatica Sinica, vol. 5, no. 5, pp. 934–945, Sep. 2018.
[3] Q. Lin, S. E. Li, X. Du, X. Zhang, H. Peng, Y. Luo, and K. Li, “Minimize the fuel consumption of connected vehicles between two red-signalized intersections in urban traffic,” IEEE Trans. Veh. Technol., vol. 67, no. 10, pp. 9060–9072, Oct. 2018.
[4] M. S. Shirazi and B. T. Morris, “Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 1, pp. 4–24, Jan. 2017.
[5] D. Hu, J. Wu, K. Tian, L. Liao, M. Xu, and Y. Du, “Urban air quality, meteorology and traffic linkages: Evidence from a sixteen-day particulate matter pollution event in december 2015, beijing,” J. Environ. Sci., vol. 59, pp. 30–38, Sep. 2017.
[6] B. Xu, X.-J. Ban, Y. Bian, W. Li, J. Wang, S.-E. Li, and K. Li, “Cooperative method of traffic signal optimization and speed control of connected and automated vehicles,” J. Environ. Sci., vol. 20, no. 4, pp. 1390–1403, 2018.
[7] J. Wu, D. Ghosal, M. Zhang, and C.-N. Chua, “Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection,” IEEE Trans. Veh. Technol., vol. 67, no. 2, pp. 896–909, Feb. 2018.
[8] W. Sun, J. Zheng, and X. H. Liu, “A capacity maximization scheme for intersection management with automated vehicles,” Transp. Res. C, Emerg. Technol., vol. 94, pp. 19–31, Sep. 2018.
[9] B. Xu, S. E. Li, Y. Bian, S. L. X. J. Ban, J. Wang, and K. Li, “Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections,” Transp. Res. C, Emerg. Technol., vol. 93, pp. 322–334, Aug. 2018.
[10] F. Gao, X. Hu, S. E. Li, K. Li, and Q. Sun, “Distributed adaptive sliding mode control of vehicular platoon with uncertain interaction topology,” IEEE Trans. Ind. Electron., vol. 65, no. 8, pp. 6352–6361, Aug. 2018.
[11] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, “Semisupervised deep reinforcement learning in support of IoT and smart city services,” IEEE Internet Things J., vol. 5, no. 2, pp. 624–635, Apr. 2018.
[12] Z.-A. Sabra, D. Gettman, R.-D. Henry, V. Nallamothu, and W. Sabra, “Balancing safety and capacity in an adaptive signal control system—Phase 1,” Federal Highway Administration, Washington, DC, USA, Tech. Rep. FHWA-HRT-10-038, 2010.
[13] A. J. Miller, “Settings for fixed-cycle traffic signals,” J. Oper. Res. Soc., vol. 14, no. 4, p. 373, Dec. 1963.
[14] Y. He, D. Sun, M. Zhao, and S. Cheng, “Cooperative driving and lane changing modeling for connected vehicles in the vicinity of traffic signals: A cyber-physical perspective,” IEEE Access, vol. 6, pp. 13891–13897, 2018.
[15] T.-H. Chang and J.-T. Lin, “Optimal signal timing for an oversaturated intersection,” Transp. Res. B, Methodol., vol. 34, no. 6, pp. 471–491, Aug. 2000.
[16] T.-L. Thorpe. (1997). Vehicle Traffic Light Control Using Sarsa. [Online]. Available: citeeseer.ist.psu.edu/thorpe97vehicle.html
[17] R.-S. Sutton, “Generalization in reinforcement learning: Successful examples using sparse coarse coding,” in Proc. Adv. Neural Inf. Process. Syst., 1996, pp. 1398–1404.
[18] B. Abdullai, R. Pringle, and G. J. Karakoulas, “Reinforcement learning for true adaptive traffic signal control,” J. Transp. Eng., vol. 129, no. 3, pp. 278–285, May 2003.
[19] V.-R. Konda and J.-N. Tsitsiklis, “Actor-critic algorithm,” in Proc. Adv. Neural Inf. Process. Syst., 2000, pp. 1008–1014.
[20] M. Aslani, M. S. Mesgari, and M. Wiering, “Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events,” Transp. Res. C, Emerg. Technol., vol. 85, pp. 732–752, Dec. 2017.
[21] L. Li, Y. Lv, and F.-Y. Wang, “Traffic signal timing via deep reinforcement learning,” IEEE/CAA J. Automatica Sinica, vol. 3, no. 3, pp. 247–254, Jul. 2016.
[22] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, “Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network,” 2017, arXiv:1705.02755. [Online]. Available: http://arxiv.org/abs/1705.02755
[23] M. Xu, J. Wu, L. Huang, R. Zhou, T. Wang, and D. Hu, “Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning,” J. Intell. Transp. Syst., vol. 24, no. 1, pp. 1–10, Jan. 2020.
[24] N. Casas, “Deep deterministic policy gradient for urban traffic light control,” 2017, arXiv:1703.09605. [Online]. Available: http://arxiv.org/abs/1703.09605
[25] W. Genders and S. Razavi, “Asynchronous n-step Q-learning adaptive traffic signal control,” J. Intell. Transp. Syst., vol. 23, no. 4, pp. 319–331, Jul. 2019.
[26] G. R. de Campos, F. D. Rossa, and A. Colombo, “Safety verification methods for human-driven vehicles at traffic intersections: Optimal driver-adaptive supervisory control,” IEEE Trans. Human-Mach. Syst., vol. 48, no. 1, pp. 72–84, Feb. 2018.
[27] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng, “Eco-departure of connected vehicles with V2X communication at signalized intersections,” IEEE Trans. Veh. Technol., vol. 64, no. 12, pp. 5439–5449, Dec. 2015.
[28] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of sumo-simulation of urban mobility,” Int. J. Adv. Syst. Meas., vol. 5, nos. 3–4, pp. 1–11, Dec. 2012.
[29] X. Liang, X. Du, G. Wang, and Z. Han, “A deep reinforcement learning network for traffic light cycle control,” IEEE Veh. Technol. Conf., no. 2, pp. 1243–1253, Feb. 2019.
[30] J. Li, Y. Si, T. Xu, and S. Jiang, “Deep convolutional neural network based ECG classification system using information fusion and one-hot encoding techniques,” Math. Problems Eng., vol. 2018, pp. 1–10, Dec. 2018.
[31] S. S. Mousavi, M. Schukat, and E. Howley, “Traffic light control using deep policy-gravity and value-function-based reinforcement learning,” IET Intell. Transp. Syst., vol. 11, no. 7, pp. 417–423, Sep. 2017.
[32] V. Mnih, A.-P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in Proc. Int. Conf. Mach. Learn., Jun. 2016, pp. 1928–1937.
[33] J. Buckman, D. Hafner, G. Tucker, E. Brevedo, and H. Lee, “Sample-efficient reinforcement learning with stochastic ensemble value expansion,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 8224–8234.
XINKE WU is currently pursuing the master’s degree in electrical engineering with the Fujian University of Technology. Her research interests are focused on reinforcement learning and deep learning. She is currently dedicating to collaborative reinforcement learning of traffic signal timing and area signal control based on special vehicles and pedestrians.

FUMIN ZOU received the Ph.D. degree in traffic information engineering and control from Central South University, Changsha, China, in 2009. He visited Texas Tech University, from December 2013 to December 2014. He is a Professor with the School of Information Science and Engineering, Fujian University of Technology (FJUT), and the Innovation Center of BeiDou Navigation and Smart Traffic of Fujian Province (ICBNST), China. His current research interests include machine learning, big data, and traffic information systems.

JENGSHYANG PAN received the B.S. degree in electronics engineering from the National Taiwan University of Science and Technology, in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from The University of Edinburgh, U.K., in 1996. He is currently the Dean of the College of Information Science and Engineering, Fujian University of Technology, and the Director of the Innovative Information Industry Research Center, Harbin Institute of Technology Shenzhen Graduate School, China. His current research interests include soft computing, information security, and signal processing. He serves on the Editorial Board of LNCS Transactions on Data Hiding and Multimedia Security, the Journal of Computers, and the Journal of Information Hiding and Multimedia Signal Processing.

QI SUN (Member, IEEE) received the Ph.D. degree in automotive engineering from the Ecole Centrale de Lille, Villeneuve-d’Ascq, France, in 2017. From 2013 to 2016, he did scientific research and completed the Ph.D. dissertation at the CRIStAL Research Center, Ecole Centrale de Lille. He currently holds a postdoctoral position at the State Key Laboratory of Automotive Safety and Energy and the Department of Automotive Engineering, Tsinghua University, Beijing, China. His research interests include intelligent vehicles, automatic driving technology, distributed control, and optimal control.

SHENGBO EBEN LI (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Tsinghua University, in 2006 and 2009, respectively. Before joining Tsinghua University, he has worked at Stanford University, the University of Michigan, and UC Berkeley. He is currently leading the Intelligent Driving Lab (iDLab), Tsinghua University. He is the author of over 100 peer-reviewed journal/conference papers and the co-inventor of over 30 patents. His active research interests include intelligent vehicles and driver assistance, reinforcement learning and optimal control, and distributed control and estimation. He also serves on the Board of Governors of the IEEE ITS Society. He was the recipient of the Best Paper Award from 14th Asian ITS, the National Award for Technological Invention of China, in 2013, the Best Paper Award from the IEEE ITS, in 2014, the Excellent Young Scholar of NSF China, in 2016, the Young Professorship of Changjiang Scholar Program, in 2016, the Tsinghua University Excellent Professorship Award, in 2017, the National Award for Progress in Science and Technology of China, in 2018, and the Distinguished Young Scholar of Beijing NSF, in 2018. He serves as an Associate Editor for the IEEE Intelligent Transportation Systems Magazine and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

MAOLIN ZHANG received the M.A. degree in transportation engineering from the Fujian University of Technology. His research interests are mainly in the fields of machine vision and artificial intelligence on transportation domain, and he is currently dedicating to vehicle video processing, and deep learning and its applications.