Received October 12, 2020, accepted October 20, 2020, date of publication October 28, 2020, date of current version November 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034419

Cooperative Control for Multi-Intersection Traffic Signal Based on Deep Reinforcement Learning and Imitation Learning

YUSEN HUO, QINGHUA TAO, AND JIANMING HU, (Member, IEEE)
Department of Automation, Tsinghua University, Beijing 100084, China

Corresponding author: Jianming Hu (hujm@mail.tsinghua.edu.cn)

This work was supported by National Key R&D Program in China (2019YFF0303102), National Natural Science Foundation of China under Grant No. 61673232 and Tsinghua University Initiative Scientific Research Program (2018Z05JDX005, 20183080016).

ABSTRACT Traffic signal control has long been considered as a critical topic in intelligent transportation systems. Most existing related methods either suffer from inefficient training or mainly focus on isolated intersections. This article aims at the cooperative control for multi-intersection traffic signal, in which a novel end-to-end learning model is established and an efficient training method is proposed analogously, which is capable of adapting to large-scale scenarios. In the proposed method, the input traffic status in multi-intersection are expressed by a tensor without information loss, which significantly reduces model complexity than using a huge matrix, since additional convolutional layers can be required to extract features from a huge matrix. For the output, a multidimensional boolean vector is employed to simplify the control policy with abiding the practical phase changing rules, and then a multi-task learning structure is used to get the cooperative policy. Instead of only using the reinforcement learning to train the model, we employ imitation learning to integrate a rule based model to do the pre-training, which greatly accelerates the convergence. Afterwards, the reinforcement learning method is adopted to continue the fine training, where proximal policy optimization algorithm is incorporated to solve the policy collapse problem in multi-dimensional output situation. Numerical experiments demonstrate the distinctive advantages of the proposed method with comparison to the efficiency and accuracy of the related state-of-the-art methods.

INDEX TERMS Deep reinforcement learning, imitation learning, multi-intersection, proximal policy optimization, tensor.

I. INTRODUCTION

Transportation systems play increasingly important roles in modern society [1], [2]. Among the key factors influencing the traffic networks, signal control on intersections plays a vital role in the operation efficiency of the system. Thanks to the development of intelligent computing technologies [3], [4], the real-time central control systems for traffic networks become more feasible. The vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications provide new approaches for traffic signal control at intersections [5], [6], in which the signal controllers can obtain accurate and dynamic information of the vehicles in real time to adaptively optimize the system. Therefore, efficient and accurate traffic signal control attracts wider attention in practice.

In fact, the traffic signal controls can be regarded as typical sequence decision problems, which well fit into the reinforcement learning (RL) framework [7]. Many researchers have been trying to use RL methods to solve the problem from different perspectives, and have shown better performance than other learning-based methods [8]–[12]. The early proposed RL based models mainly used the Q-learning method to convert the traffic states to discrete values, but they suffered from the curse of dimensionality due to the storage of large Q-table [9], [11]. Therefore, it often makes a rough division of traffic states at the cost of information loss.

With the development of deep learning, deep RL has also been studied. Deep RL integrates traditional RL with the deep neural network based value function approximation methods,

The associate editor coordinating the review of this manuscript and approving it for publication was Chao-Yang Chen.

and is capable of conquering the problem of dimensionality curse. Many traffic signal control models adopted such methods, which transformed the input traffic states into the values of different control actions where the highest values were chosen as the actions for each decision [10], [12]–[16]. In [14], it transformed real-time traffic states into position matrices and speed matrices, and adopted convolutional networks to automatically extract traffic features. These methods solve the problem of information loss in traffic states, but they only deal with the isolated intersections and also ignore the problems of spatial coupling.

Recently, multi-intersection modeling problems have attracted increasing attentions [15], [17]–[19]. The control model of multi-intersections currently faces three major challenges. Firstly, massive real-time traffic data are now collected and should be fused in an appropriate way. Secondly, cooperation mechanisms of the intersections are required to improve performance since traffic chaos can be caused due to the interference among each other. Thirdly, for the learning based models, training complexity remains intractable when they are deployed in practice. In [15], the states sharing multiple intersections were tackled by concatenating the position matrices of multiple intersections, but they concatenated the matrices just by simple splicing, which greatly increased the dimensionality of the inputs and was difficult to extend. In [18], a control model was proposed to integrate multi-agent deep RL into a huge graph neural network. The model can be deployed in a network with thousands of intersections. However, all the agents invovled were optimized without considering the interaction with each other, namely lacking effective cooperation mechanisms. Moreover, most of the control models require thousands of training iterations, leading to a infeasibility for deployment.

To overcome the challenges, this article establishes a novel end-to-end cooperative control model for multi-intersection traffic signal under the V2X based traffic systems [20], which is named as deep reinforcement learning and imitation learning based multi-intersection signal control model (DRI-MSC). For the fusion method, the proposed DRI-MSC model constructs a tensor to represent the traffic information, where each matrix describes one intersection. Compared with the method using a huge matrix to cover all the traffic information of the multi-intersection, the DRI-MSC model reduces dimensionality with computational intractability and avoids information loss. To handle the the cooperative control problem, a learning based cooperative method is proposed, which uses a multi-task learning structure with multi-dimensional output to directly transform the input tensor to a cooperative policy, and employs the proximal policy optimization (PPO) algorithm to reduce the impact of policy collapse caused by the multi-dimensional output [21]–[23]. Compared with the other rule based cooperative policies, our method can avoid local optima with better extendability, scalability and efficiency for large-scale scenarios.

For the training complexity, the proposed DRI-MSC model first introduces the supervised imitation learning method to initialize the training with a satisfactory solution, which greatly accelerates the poor convergence in deep RL based methods. Then, RL method is employed to fine tune the model. The combined training method omits a large number of invalid searches to greatly improve the convergence rate and provides more possibilities to skip the local optima. In addition, the proposed method also alleviates the oscillation amplitude of the policy in convergence. Numerical experiments verify the effectiveness of the proposed model regarding efficiency and accuracy, which illuminates the practical applications for multi-intersection scenarios in the future.

The remainder of the paper is arranged as follows. Section II introduces the background of modeling and training. Section III presents details of the proposed DRI-MSC model. Numerical experiments based on different traffic flows are conducted in Section IV, and Section V ends the paper with conclusion.

II. BACKGROUND

A. TRAFFIC SIGNAL CONTROL MODEL

Intersections are considered as the bottlenecks of traffic network, where vehicles gather and dissipate. The most common intersections are the 4-way and 3-way (T-junction) type cross road. In the modern traffic systems, there are multiple intersections in nearby areas, which coordinate to control the traffic networks. Fig. 1 shows the traffic networks with single intersection and 4 intersections.

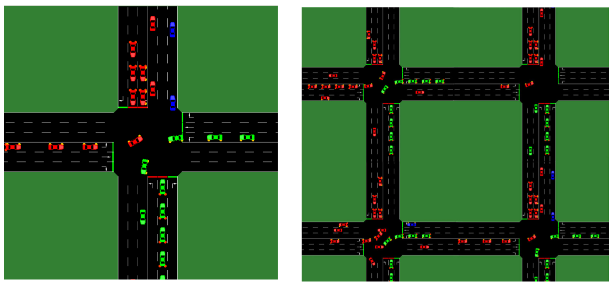


FIGURE 1. The left sub-figure shows a traffic network with single intersection, while the right sub-figure with 4 intersections.

In order to model the multi-intersection scenarios, the input features, the model output, the collaborative control policy and the evaluation indexes should be taken into account. Typically, features such as traffic flow size, average vehicle speed and lane occupancy are often used to reflect traffic states. Generally, these features are represented as a matrix, such as position matrix for specific description to the traffic states [14], which inspires the model input of this article.

The output of the control model can be converted to different policies of traffic lights, which control the traffic flow directly. Typically, control model either output the duration of green phases for different directions, or just output the decision whether the green phase change to the next directions [10], [14], [24]. We name them as the phase based output.

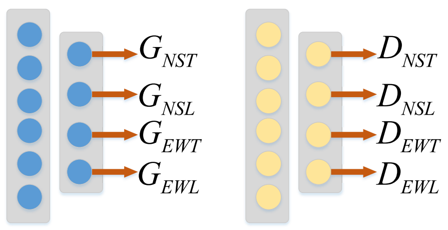


FIGURE 2. The left is for the phase output, and the right for the phase duration, where "G' and "D" represent the decision and duration of each green phase respectively. For the first two digits in the subscript, "N", "S", "E" and "W" represent the north, south, east and west respectively, while the last digit "T" and "L" denote the through and left-turn direction.

For the former one which is shown in the left of Fig. 2, the sequence of phases is disrupted which is forbidden in the real world. For the latter one shown in the left, the sequence of phases can be guaranteed at the cost of the hysteresis of the timing policy.

Typically, In order to output collaborative control policy of multi-intersection, human-designed rules are used to modify the output of sub-agents to get the cooperative output, which is difficult to design and prone to fall into the local optima.

For the objective, indexes such as queue length, waiting time of vehicles, fuel consumption and accumulative delay are often used to quantify the traffic efficiency. Then programming methods and learning-based methods are used to calculate the optimal value of the these indexes. For learning-based methods, which are applied in this article, RL are mostly used, and the indexes are converted into rewards and maximized through learning process.

B. REINFORCEMENT LEARNING

RL is an adaptive learning framework whose idea comes from biological mechanism, and it is a method for agents to constantly change and optimize their policies based on past experience [7]. The main characteristic of RL is to learn good policies through exploration by trial-and-error and exploitation by the existing experience. In the decision-making process, RL based models map each state s_t to an action a_t, and then receive rewards from environmental feedback to continuously optimize its policies accordingly. The goal of RL is to maximize the total future reward, which can be measured with a value function $v_\pi(s_t)$, regarding the cumulative expectation of the rewards under policy π, i.e.,

$$v_\pi(s_t) = E_{\tau\sim\pi}\left[\sum_{k=t}^{T}\gamma^k R_k\right], \tag{1}$$

where t represents the current time-step, τ denotes the history trajectory, R_k is the reward of one time-step k, and T means the terminal time-step. In addition, the future reward at each time-step k can have a decay with a certain rate $0 < \gamma < 1$, meaning that the states far away from the current time-step have lower weights. For the traffic control problem considered in this article, a larger γ gives higher weight to long term traffic efficiency, and herein can be more useful to global optimization.

RL methods can be mainly categorized into two types. One is the value-based approach, which makes decisions by calculating the values of different selected actions in the current state. Deep Q net (DQN) is a typical instance for this idea, but this methods is not suitable for the multi-dimensional output model. Another type is the policy-based approach that maps states to policies, and one representative method is the policy gradient algorithm, which uses a neural network to straightforwardly output multi-dimensional actions [25]. The latter is suitable to control multiple signal controllers simultaneously and more feasible for large scale problems.

1) POLICY GRADIENT AND OPTIMIZATION

For a policy-based model, the actions are mapped from the parameter space, and the environment then maps the actions to the rewards. Combined with (1), we can bridge a functional relationship between parameters θ and value function $v_\pi(s_t)$. The gradient of values with respect to θ can be calculated by back-propagation algorithm, and then the gradient descent method can be used to update the parameters [26]. Analogously, the policy gradient $\nabla_\theta v_\pi(s_t)$ can be calculated as

$$\nabla_\theta v_\pi(s_t) = E_{s_t\sim d^\pi, a_t\sim\pi}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) A^\pi(s_t, a_t)], \tag{2}$$

where d^π denotes the distribution of the s_t under policy π, and $A^\pi(s_t, a_t)$ is called the advantage function [27], which represents the additional value under action a compared with the expected value of s_t.

2) ACTOR CRITIC

Practically, $A^\pi(s_t, a_t)$ is hard to calculate directly, and these value approximation methods using learning based models to approximate the $A^\pi(s_t, a_t)$ can be taken as valid solutions. The method which combines policy gradient with value approximator is called actor critic. In this article, actor critic is applied to improve the convergence rate of the model, where a neural network is used as an approximator and is called value network.

C. IMITATION LEARNING

Imitation learning utilizes machine learning models to imitate behaviors of the experts or other models. When the model can accurately predict the behaviors, we only need to make decisions based on the results predicted by such model. Moreover, imitation accuracy can also be used to judge the feature extraction and model ability, which means low accuracy may be caused by defective feature extraction method or insufficient model structures.

A typical method of imitation learning is to train classifiers or regressors to predict the behaviors of the experts based on input states and the observed output behaviors. However, the behaviors of learners also affect the future input observations and states during the implementation of policies, which results in the violation of the I.I.D. (independent and identically distributed) hypothesis of the methods. In practice, once a learner makes a mistake, it can encounter observation data which are completely different from the presentation of experts, leading to the compound of errors.

In order to solve this problem, trajectory data should also be added to the training data dynamically. Dataset Aggregation (DAGGER) collects trajectory data at the same time when model makes decisions, and label the new collected data by experts in real time [28]. Then the training set can cover most of possible scenes through repeated training and decision making process, and the model can perform as an expert. In this article, we use DAGGER to pre-train our model with a simplified rule based model as labels.

III. THE DRI-MSC MODEL

In this section, we introduce the proposed DRI-MSC model regarding its structure for modeling the multi-intersection control problems and the corresponding training algorithm.

A. STRUCTURE OF THE DRI-MSC MODEL

1) TENSOR-BASED INPUT

The traffic state information includes the number of intersections in the region, the number of cars in each lane, the distance between traffic flows intersections, etc. In order to avoid information loss in feature extraction and maintain the feature matrices tractable, we use tensors to represent the traffic state at each moment.

Under the V2X based traffic systems, it is reasonable to get the full scene information of traffic states for a specific intersection and time-step. In each lane, the right side is the direction close to the intersection. Thus the lanes with vehicles towards right are the entry lanes, which lead to intersections, and the lanes with vehicles facing left are the exit lanes, which depart from intersections. Fig. 3 shows a typical intersection with 3 lanes at each direction, and there are altogether 24 lanes from top to bottom. We then extract features by transferring the traffic states into a cellular matrix [14]. Specifically, each lane is cut into small blocks at an interval of 5 meters, and every block is assigned with a boolean value to present the presence and absence of a standard vehicle, which is roughly the same as a car. All the values in the blocks make up of the cellular matrix, where the coordinates of each "1" reflect the actual position of the corresponding standard vehicles. In this setting, large vehicles would occupy more than two blocks, and be converted into more standard vehicles.

For multiple intersections, we superimpose cellular matrices of different intersections to form a tensor, which is defined as

$$s_t \in R^{I \times J \times K}, \tag{3}$$

where K represents the number of blocks in each lane, J is the number of lanes and I denotes the number of intersections. The element (i, j, k) of tensor s_t is denoted by $s_t^{(i,j,k)} \in \{0, 1\}$, in which $s_t^{(i,j,k)} = 1$ means the presence of a vehicle in the position (i, j, k). Moreover, intersections are in the position

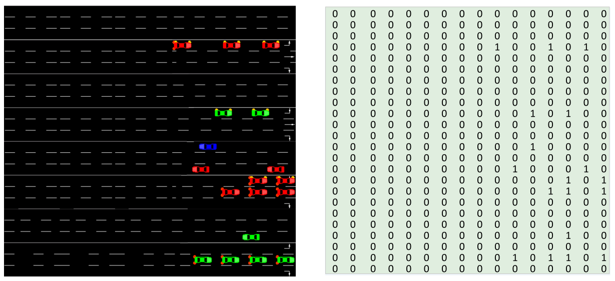


FIGURE 3. The matrix for traffic states in a single intersection.

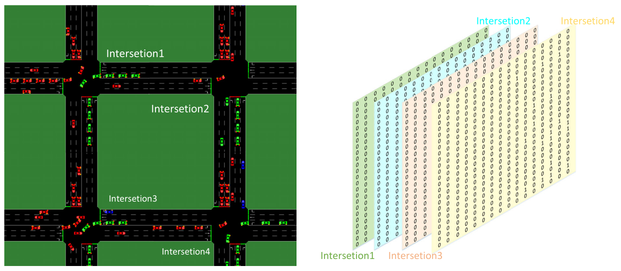


FIGURE 4. The tensor for traffic states in 4 intersections.

of (i, j, K). For example, for the 4 intersections in Fig. 4, we first get the cellular matrices for each intersection, then superimpose the matrices to form a tensor. On such basis, two convolutional layers are used to automatically extract traffic features.

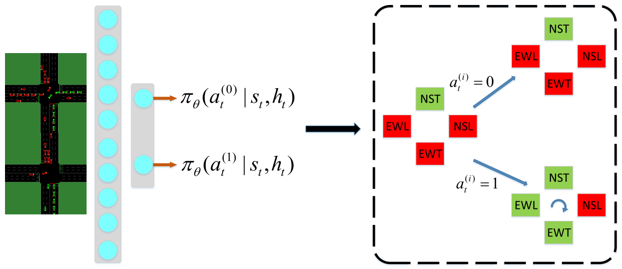
In this article, we use a matrix to represent the phase information of the whole traffic network. which is defined as

$$h_t \in R^{I \times M}, \tag{4}$$

where I is the number of intersections and M represents the phase number of an intersection. The rows of h_t represent the intersections, and the columns represent the indices of phases. Element (i, m) stands for the state of the m-th green phase of the intersection i, and is denoted by $h_t^{(i,m)} \in \{0, 1\}$. When it is the current phase, we have $h_t^{(i,m)} = 1$. Moreover, M is equal to the phase number of the intersection with the most directions. For example, for the typical four-phase intersections involved in the experiment of this article, $M = 4$. If a T-junction exist, "$0''$" is added for unified representation. We use a dense layer to embed the phase matrix with the traffic features output by convolution layers, and then the embedded features are transformed to outputs.

2) OUTPUT

Although the number of phases is determined by the number of directions, the sequence of phases are required to be fixed, where we only need to decide if the green phase should be changed or not. Based on this, a binary action is sufficient to control the signal controller. Therefore, we propose a boolean variable to represent the binary action which is whether to change the green phase or not. We name this pattern as the



$\pi_\theta(a_t^{(0)} \mid s_t, h_t)$
$\pi_\theta(a_t^{(1)} \mid s_t, h_t)$
NST
EWL
NSL
EWT
$a_t^{(i)}=0$
$a_t^{(i)}=1$

FIGURE 5. The left shows the output of DRI-MSC model, where $\pi_\theta(a_t^{(0)}|s_t, h_t)$ and $\pi_\theta(a_t^{(1)}|s_t, h_t)$ stands for the the probability of $a_t^{(0)}$ and $a_t^{(1)}$. The right shows the phase order, where the green color stands for the green phase, the right color stands for the red phase, and the green phase changes clockwise when $a_t^{(i)} = 1$.

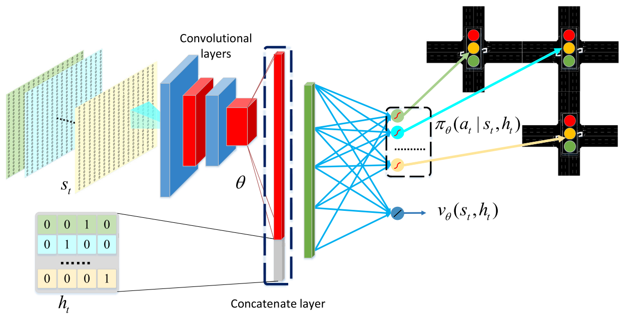


FIGURE 6. The decision structure of the DRI-MSC model.

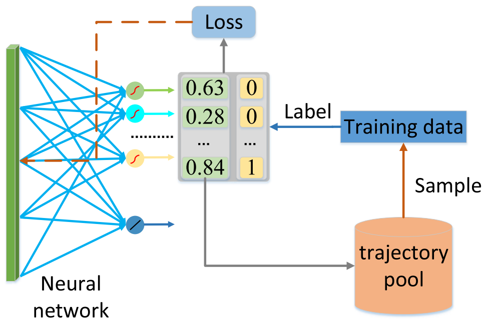


FIGURE 7. The imitation learning process.

order-based output. Specifically, as shown in the right of Fig. 5, the order-based vector of multi-dimensional actions is defined as

$$\pi_\theta(a_t|s_t, h_t) = f(s_t, h_t|\theta), \tag{5}$$

where $f(\cdot)$ stands for the neural network, and a_t is defined as the switching vector. The ith element of a_t decide whether the green phase change direction for the ith intersection, and is denoted by $a_t^{(i)} \in \{0, 1\}$, where $a_t^{(i)} = 1$ means the green phase of the ith intersection changing to the next direction.

We also have $\pi_\theta(a_t|s_t, h_t) \in R^I$, where I represents the number of intersections and θ stands for the parameters of the neural network. The ith element of $\pi_\theta(a_t|s_t, h_t)$ means the probability of $a_t^{(i)}$, and is denoted by $\pi_\theta(a_t^{(i)}|s_t, h_t) \in [0, 1]$, and $\pi_\theta(a_t^{(i)} = 0|s_t, h_t) + \pi_\theta(a_t^{(i)} = 1|s_t, h_t) = 1$. The dimension of the actions represents the policy for the corresponding intersections. Compared to the traditional phase based output, the output of DRI-MSC model abides the practical phase changing rules, which provides a more simplified and reasonable representation.

3) MULTI-TASK LEARNING STRUCTURE

In robotic learning, the joints of the robots cooperate to complete complex tasks such as jumps and somersaults completely through learning [22]. We extend this idea to the multi-intersection traffic signal control model. Therefore, the DRI-MSC model get the collaborative policies through end-to-end multi-task learning, which is shown in Fig. 6.

The main idea of this architecture is to define a unified optimization objective which regards the policy of each signal controller as a sub-task, and use a neural network with shared parameters in multi-task architecture to output the collaborative action of each task [29]. Finally, a learning based method is used to train the neural network. Compared to the rule based collaborative policy, the proposed method can break through the limitation of human designing and better approximate the optimal solution.

The multi-task learning architecture in the proposed DRI-MSC model combines the policy gradient and temporal-difference errors into one, while the existing works use isolated value network and policy network and separates the total loss into two parts [24], [30]. The DRI-MSC model can reduce the oscillation and instability while providing a more refined structure and more stable convergence. The DRI-MSC model output probability vectors $\pi_\theta(a_t|s_t, h_t)$ and values $v_\theta(s_t, h_t)$ at the same time, where $\pi_\theta(a_t|s_t, h_t)$ is the control policy and $v_\theta(s_t, h_t)$ is the value of state s_t.

B. TRAINING METHOD

1) PRE-TRAINING WITH IMITATION LEARNING

Since the supervised learning is normally faster to converge, we pre-train the model to guarantee a warm start point for RL by imitating the policy of a rule based model. We use DAGGER algorithm to pre-train the model, and the learning process is shown in Fig. 7.

We collect the decision trajectory data into a trajectory pool while the model interacts with the environment, and then randomly sample batches of data from the trajectory pool as training samples which are labeled by a rule based multi-intersection signal control model. Therefore, a good baseline control policy can be provided. In this article, we use the following formula to represent the model,

$$y^{(i)} = \begin{cases} 0 & \sum_{m=1}^{(M-1)} N_m^{(i)} \cdot \beta - N_0^{(i)} \leq 0 \\ 1 & \sum_{m=1}^{(M-1)} N_m^{(i)} \cdot \beta - N_0^{(i)} > 0 \end{cases} \tag{6}$$

where $y^{(i)}$ represents the label of the ith dimension, which is also the rule-based policy corresponding to the traffic state of the intersection i. Besides, $N_0^{(i)}$ indicates the number of low-speed vehicles in lane i with green phase, and $\sum_{m=1}^{(M-1)} N_0^{(i)}$ indicates the number of low-speed vehicles in the lanes with red phase, and M is the phase number. Noticing that the density of traffic flow can be indirectly reflected by the speed of the vehicles, we could measure the congestion via the number of halting and slow-moving vehicles. In this article, we call these vehicles the speed-limited vehicles, and use the number of speed-limited vehicles to roughly reflect the density and congestion of traffic state. The core of (6) is that if the direction with green phase is much more crowded than other directions, then we keep the existing phase, otherwise it switches to the next phase. The advantage of such settings lies on the fact that the policy model is very small and is easy to calculate and achieve a certain result. Moreover, the velocity bound of the low-speed vehicles needs to be determined according to the actual situation of the road such as road grades and maximum speed limit.

The loss of the imitation learning model is supposed to reflect all the intersections, thus we define the global loss as the sum of errors in each dimension, which is formulated as

$$E = \sum_{t=0}^{T-1} \sum_{i=0}^{I-1} e_t^{(i)} + c \left\| \theta \right\|^2, \tag{7}$$

where $e_t^{(i)} = y_t^{(i)} \ln(\pi_\theta(a_t^{(i)}|s_t, h_t)) + (1 - y_t^{(i)}) \ln(1 - \pi_\theta(a_t^{(i)}|s_t, h_t))$ and $y_t^{(i)}$ represents the label of the first dimension. T means the terminal time-step defined in (1). E is designed to measure the similarity between the model outputs and the sample labels. In addition, we added l_2 norm penalty to avoid over-fitting. Then we employ the stochastic gradient descent algorithm for the optimization.

2) FINE-TUNING WITH RL

After the pre-training to accelerate the convergence, RL method is then used to fine tune the model.

For a RL based model, the reward function gives a quantitative description of the optimization objective and a evaluation criterion for policies and actions. In a traffic network, since adjacent intersections influence each other, individual optimization on each intersection can lead to the prisoner's dilemma for the whole network [31]. Previous works either use the change in cumulative vehicle delay, or use a simplified queue length as reward [12], [14]. However, delay is a lagging indicator and convergence rate can be negatively impacted. The coming flow and speed-limited vehicles can be ignored when employing simplified queue length of halting vehicles. Due to the complexity of traffic states and the mutual influencing mechanism among intersections, we describe the total spatial objective by setting the reward function as the change of numbers of speed-limited vehicles in all lanes of the entire traffic network,

$$R_t = \sum_i (N_t^{(i)} - N_{t+1}^{(i)}), \tag{8}$$

where $N_t^{(i)}$ represents the number of speed-limited vehicles in lane i at time t. In a short period of time, R_t can be noised since it fluctuates with the change of the phases, whereas in a long term, noise can be cancelled by each other, and thus the reward can reflect the effect of the policies. When traffic conditions improve, the number of speed-limited vehicles decreases and the model receives more higher rewards and vice verse.

For the actor critic method applied in this article, $v_\theta(s_t, h_t)$ is used to estimate the real value $v_\pi(s_t, h_t)$ defined in (1), and $A_n(s_t, h_t, a_t)$ is defined as

$$A_n(s_t, h_t, a_t) = \sum_{k=1}^{n} (R_{t+k} + \gamma^k v_\theta(s_{t+n}, h_{t+n})) - v_\theta(s_t, h_t), \tag{9}$$

which is used to estimate $A^\pi(s_t, h_t, a_t)$ in (2) based on bellman equation [32]. In this article, different time steps $A_1(s_t, h_t, a_t), A_2(s_t, h_t, a_t), \cdots A_N(s_t, h_t, a_t)$ are used to make a proper trade off between short-term and long-term rewards and N stands for the max step [33]. Moreover, $A_n(s_t, h_t, a_t)$ can be regarded as temporal-difference error, which can be used to calculate the error of the value network [34]. Then, the objective function of the estimated value $v_\theta(s_t, h_t)$ is shown as follows

$$L(v_\theta(s_t, h_t)) = \frac{1}{N} \sum_{n=1}^{N} (A_n(s_t, h_t, a_t)). \tag{10}$$

For the multi-dimensional action model in this article, the convergence remains a problem due to the mismatch between parameter space and action space and the high coupling between policies and environment [35]. A solution to this problem is to limit the update magnitude of a policy by adding constraints [36]–[38]. In this article, we use PPO method whose main idea is to strict the magnitudes of the parameters updates by truncating the policies. The target function corresponding to this method can be expressed as follows

$$\begin{aligned} L_{\pi_{\theta'}}(\pi_\theta) = E_{\tau \sim \pi_{\theta'}}[\min(A_n(s_t, h_t, a_t), \\ \mathrm{clip}(r_t^{\pi_{\theta'}}(\pi_\theta), 1-\varepsilon, 1+\varepsilon) A_n(s_t, h_t, a_t))], \end{aligned} \tag{11}$$

with

$$r_t^{\pi_{\theta'}}(\pi_\theta) = \frac{\pi_\theta(a_t \mid s_t, h_t)}{\pi_{\theta'}(a_t \mid s_t, h_t)}, \tag{12}$$

where π_θ and $\pi_{\theta'}$ stands for the policy based on θ and θ', and $r_t^{\pi_{\theta'}}(\pi_\theta)$ is the ratio of the new policy to the old policy. Moreover, ε is the threshold of policy change, and $\mathrm{clip}(r_t^{\pi_{\theta'}}(\pi_\theta), 1 - \varepsilon, 1 + \varepsilon)$ means clipping the probability ratio $r_t^{\pi_\theta}(\pi_{\theta'})$ to a range of $[1 - \varepsilon, 1 + \varepsilon]$.

Fig.8 gives the illustration of the RL process used in this article. The total objective function $L(\theta)$ is as

$$L(\theta) = L_{\pi_\theta}(\pi_{\theta'})) - \alpha_1 L(v_\theta) + \alpha_2 S(\pi_\theta), \tag{13}$$

where the policy function $L_{\pi_\theta}(\pi_{\theta'})$ can be computed by (11) and value function $L(v_\theta)$ can be computed by (9). $S(\pi_\theta)$ is the entropy of the policy π_θ, which is a soft RL based regularization method [39]. The main idea of soft RL is to succeed at the task while acting as randomly as possible.

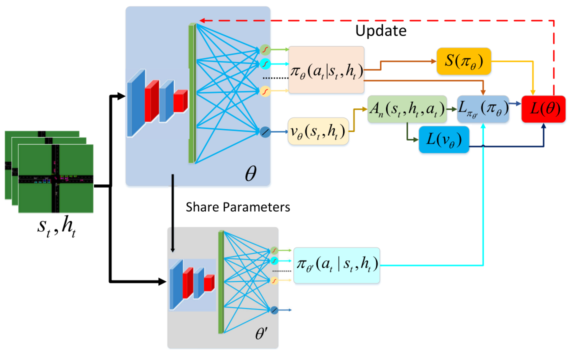


FIGURE 8. The reinforcement learning process.

Experiments show that this method can improve the stability and scalability of RL model [40], [41].

Then the new policy π_θ of this iteration (or the old policy $\pi_{\theta'}$ of the next iteration) can be obtained by maximizing $L(\theta)$:

$$\pi_{\theta'} \leftarrow \underset{\pi_\theta}{argmax}(L(\theta)) \tag{14}$$

Fig. 8. shows the complete parameter update process, where the policy is improved iteratively.

3) CO-TRAINING

The details of the co-training process are shown in Algorithm 1. In the first stage of training, we use the accuracy Acc defined as

$$Acc = \frac{\sum_{t=0}^{T-1}\sum_{i=0}^{I-1}\left|a_t^{(i)} - y_t^{(i)}\right|}{T+I}. \tag{15}$$

to evaluate the model output. Since imitation learning reduces the search space simultaneously, exorbitant Acc can lead to a negative impact on RL performance. Therefore, we introduce a early stop mechanism, which sets a threshold ξ to control the imitation accuracy. The imitation learning process ends when Acc reaches ξ, and then the reinforcement learning process start.

Algorithm 1 DRI-MSC Co-Training Algorithm

```
Randomly initialize new policy network $\pi_\theta$, old policy network $\pi_{\theta'}$ with weights $\theta$ and $\theta'$, the traffic state $s_1$ and the trajectory pool $D$.
repeat
    for $t = 1 \to T$ do
        Choose action $a_t$ based on policy $\pi_\theta$;
        Execute $a_t$ and observe the next state $s_{t+1}$,$h_{t+1}$;
        Store $s_{t+1}$, $h_{t+1}$ in $D$;
        Sample a random minibatch of trajectory data $s_{r1}, h_{r1}, s_{r2}, h_{r2}, \cdots, s_{rN_b}, h_{rN_b}$;
        Compute lables $y_{r1}, y_{r2}, \cdots, y_{rN_b}$;
        Train the new policy network with respect to $\theta$ for $m$ iterations.
    end for
until $Acc > \xi$
for $n = 1 \to N_{rl}$ do
    for $t = 1 \to T$ do
        Choose action $a_t$ based on policy $\pi_\theta$;
        Execute $a_t$ and observe the next state $s_{t+1}$, $h_{t+1}$;
        Store transition $(s_t, h_t, a_t, r_t, s_{t+1}, h_{t+1})$ in buffer.
        for every $N$ steps do
            Update the weights of old policy network $\theta'$ with $\theta$

            Compute advantage function $A_1(s_t, h_t, a_t), A_2(s_t, h_t, a_t), \cdots, A_N(s_t, h_t, a_t)$
            Update the new policy network by maximizing $L(\theta)$
            Clear buffer.
        end for
    end for
end for
```

IV. NUMERICAL EXPERIMENT

In this section, a series of numerical experiments are conducted to evaluate the proposed DRI-MSC model from different aspects. Under different scenes and flows, comparisons with related state-of-art methods are presented to illustrate the efficacy and advantages of the proposed DRI-MSC model in tackling large scale control problem for multi-intersection traffic signal. Then detailed experiments and analysis towards the specific policies in DRI-MSC model are then discussed to give further insights.

A. THE BASIC SETTINGS AND ASSUMPTIONS

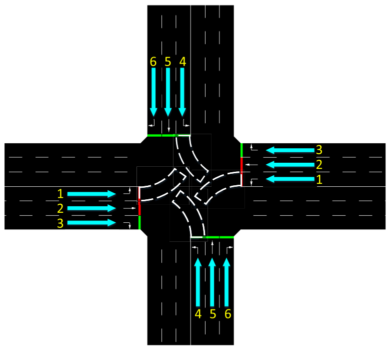
The simulation of urban mobility (SUMO) platform is taken as the experimental environment, in which we standardize the experimental environment and objects. Specifically, all intersections are typical 4-way, the number of lanes is set as 3 in each direction, the width of each lane as 3.5 meters (m), the maximum speed of the lane as 50 kilometer per hour (km/h), and the length of road as 500m in all intersections. For the intersections, as shown in Fig. 9a, the rightmost lanes are right-turn lanes whose phases are always green, and the middle lanes are through lanes.

For the phase setting, the phase order is first north-south straight, then north-south left, then east-west straight, and last east-west left. In order to ensure that the model is similar to the real world and the traffic safety is guaranteed. Amber phase with the duration of one time-step is inserted into a phase change, and the model outputs a policy at each time-step. Moreover, we set the minimum duration of green phase in any direction to be no less than 6s to ensure the minimum passing time of vehicles, and the maximum duration to be no more than 30s to avoid excessive waiting time.

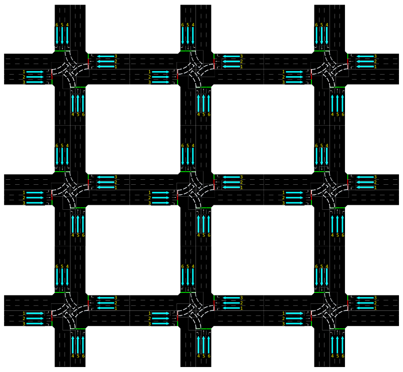
In the experiments involved in this article, we just assume that the permeability of V2X devices is 100%, and information of each car in all lanes can be obtained, all the traffic flows are composed of cars which abide the traffic rules, and the traffic flow is uniformly distributed in the system. In addition, pedestrians are not considered, and a left-turn

TABLE 1. Illustration of experimental flows.

Flow	1 (n/S)	2 (n/S)	3 (n/S)	4 (n/S)	5 (n/S)	6 (n/S)	Time (S)	Total
Low	600	600	600	600	600	600	1	3600
Middle	800	800	800	800	800	800	1	4800
High	1000	1000	1000	1000	1000	1000	1	6000
	600	600	600	600	600	600	1/3	1200
Mutable	900	900	900	900	900	900	1/3	1600
	720	720	720	720	720	720	1/3	1440
	600	900	600	900	600	600	1/3	1400
Unbalanced	900	600	600	600	900	600	1/3	1400
	900	900	600	600	900	600	1/3	1400



(a) Isolated intersection.



(b) Traffic network with 9 intersections.

FIGURE 9. Illustration of experimental scenes.

waiting area is set. For the experiment setting, we set the duration of each simulation episode as 4000 seconds (s). All the experiments are conducted on Intel(R) Core(TM) i7-6900k CPU @3.20GHz with NVIDIA GTX TITAN Xp Pascal.

Fig. 9 shows the experimental scenes in the paper. Table 1 describes the flow data in the single-intersection experiment, where "S" represents one simulation episode, and "n/S" is short for the number of vehicles in every simulation episodes. The direction of flows in Fig. 9a is indicated by arrows indexed with numbers. For example, the arrow "1" indicates the direction of route 1, and represents the flow in the east-west direction. Without specific notation, the traffic data at multiple intersections are all low flows.

All the hyper-parameters of the neural network of DRI-MSC model are show in Table 2, where "CONV",

TABLE 2. Parameters of the neural network architecture.

Layers	Kernel Size	Number	Activation
CONV 1	5×5	32	Relu
MP 1	1×2	1	Linear
CONV 2	3×3	64	Relu
MP 2	2×2	1	Linear
FC	$--$	500	Relu
FC	$--$	$Num + 1$	Sigmoid/Linear

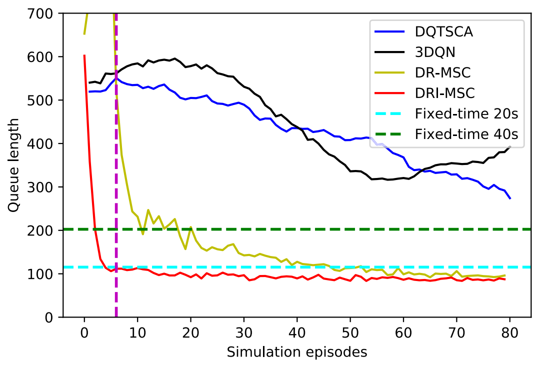
"MP" and "FC" represent convolution layer, max pooling layer and fully-connected layer, respectively. The DRI-MSC model contains two convolution layers to extract features from the input tensors and two pooling layers to reduce dimensionality [42]. Besides, we concatenate the features with phase information, which is maped to the output of the model with a fully connection layer.

For a network composed of *Num* intersections, the number of neurons in the output layer is $Num+1$, in which the sigmoid activation function maps the probability to control the traffic light at each intersection and the value of the traffic state $v(s_t)$ is also outputted additionally to compute policy gradient.

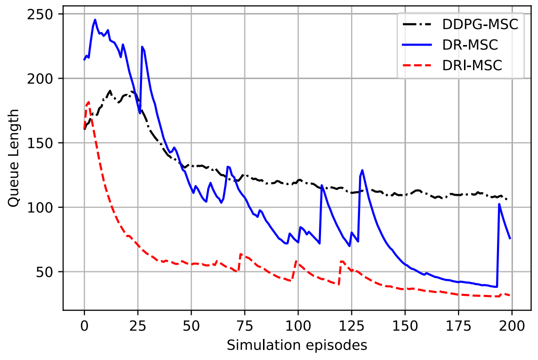
B. TRAINING OF DRI-MSC MODEL

For the imitation learning process, we set the coefficient c of (7) as 10^{-4}. In Algorithm 1, after one simulation episode (4000s), $m = 500$ training iterations are conducted with batch size $N_b = 100$, and the threshold $\xi = 0.9$ when facing the isolated intersection environment. In the multi-intersection environment, we set $\xi = 0.7$ to encourage more exploration. For the RL process, we set γ in (1) as 0.6. In equation (13), α_2 is set to 0.1 to control the convergence speed and balance exploration with exploitation. The coefficient α_1 is then employed to balance the priority of policy objective and value objective. We found in the experiments that larger and smaller α_1 both have a negative influence on convergence, and $\alpha_1 = 1$ is set as a satisfactory trade off. In addition, the vehicles with a speed lower than 30km/h are defined as a speed-limited vehicle, and we set $\beta = 0.13$.

For imitation learning based model, we use imitation loss and accuracy to the rule based model as evaluation indexes. For RL based model, we use entropy as value loss to evaluate the convergence. For a traffic control model, indexes such as queue length, average waiting time, average speed and energy consumption per vehicle can reflect the efficiency, and we adopt average phase duration as another evaluation index.

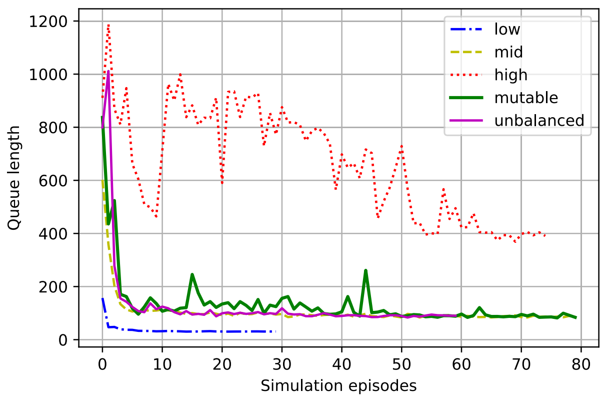


(a) The comparison of models for isolated intersections.

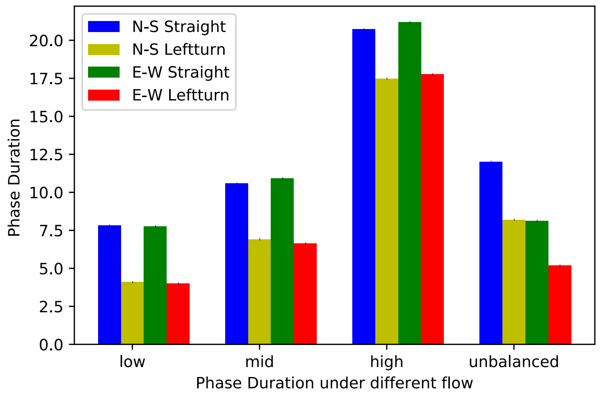


(b) The comparison of models for 4-intersection.

FIGURE 10. The comparison with other stare-of-art methods.



(a) Queue length of different flow.



(b) Queue length of different intersection numbers.

FIGURE 11. The phase duration under different flows, in which "N", "S", "E" and "W" represent the north, south, east and west respectively.

C. EXPERIMENTAL RESULTS AND ANALYSIS

We first compare the convergence of DRI-MSC model with other methods, where the deep Q-network traffic signal control agent (DQTSCA) model, double dueling deep Q-network (3DQN) model and two fixed-time control policies are introduced for the isolated intersection [10], [14]. For multi-intersection, we compare the proposed DRI-MSC model with the deep deterministic policy gradient model based multi-intersection signal control (DDPG-MSC), which is a benchmark deep RL model applicable for multi-dimensional control. For the DDPG based model, we set the capacity of the experience replay to 10000, and softly assign the parameters with a rate of 0.01. All the other parameters are the same as DRI-MSC model. Since DQTSCA model and 3DQN are only based on deep RL, we also compare the DRI-MSC model without imitation learning (DR-MSC) to test the effect of the RL and imitation learning methods in our proposed model. Fig. 10a gives the comparison results.

Fig. 10a shows that the queue length of DQTSCA and 3DQN models need hundreds of simulation episodes to converge to the level of fix-time policies, while the DRI-MSC model and the DR-MSC model only needs several episodes to converge by contrast. For the 4-intersection experiment in Fig 10b, it shows that the DRI-MSC model converges faster than the DDPG-MSC model. The DR-MSC model has a similar convergence speed with that of the DDPG-MSC model at the initial epochs, and then DR-MSC model also converges faster with epochs going on. Except the convergence speed, we can also see that the proposed DRI-MSC model can reach a better result, i.e., a shorter queue length. Fig. 10a and Fig 10b verify that the proposed DRI-MSC model outperforms the given methods both in convergence speed and final performance. Besides, Fig 10b also illustrates that imitation learning method accelerates the converging process in the proposed DRI-MSC model.

After verifying the advantages of the DRI-MSC model, we then evaluate it overall performance under different conditions and from different aspects, together with corresponding analysis. We first evaluate the DRI-MSC model in Fig 11 with different flows given shown in Table 1.

Fig. 11a illustrates that the DRI-MSC model converges in all the flows given. We can see that the model converges slower when the traffic become heavier, which is reasonable. Under the low flow condition, the model can converge to a good point within two episodes of simulation. Except high flow, the model converges to a good point within less than 7 hours theoretically when deployed in the practice. Therefore, the DRI-MSC model based policy is more acceptable and applicable when compared with other RL models,

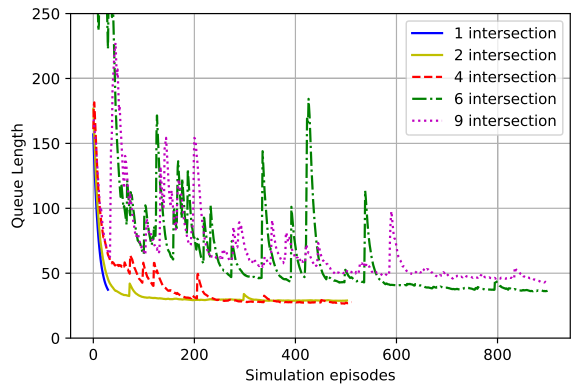


FIGURE 12. **Converging curves of DRI-MSC model.**

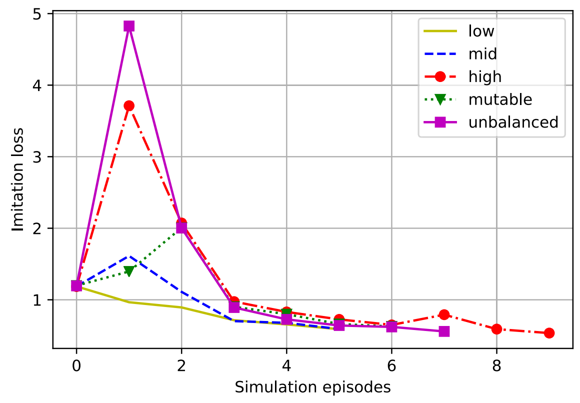
since the ordinary RL models require up to months of training cost. Moreover, mutable flow and asymmetric flow do not significantly increase the training time under the condition of a certain traffic intensity, which shows good adaptability of the proposed DRI-MSC model to the common traffic flow changes in practice. Therefore, we can conclude that the DRI-MSC model converge fast with low fluctuation in these representative conditions.

Fig. 11b shows the average phase duration of the policies. We can see that the average duration of phases for the through lanes is always longer than that of the phases for left-turn directions since the left-turn waiting areas help to reduce the passing time. Moreover, the duration of the phases keeps increasing with the increase of traffic flow, since heavier traffic requires longer phase duration. In addition, when the traffic of north-south through direction and north-south left-turn direction is heavier than east-west through direction and east-west left-turn direction, the corresponding phase duration also changes accordingly. Therefore, the DRI-MSC model is proved to be capable of optimizing the efficiency of traffic conditions. Furthermore, the DRI-MSC policy can also be regarded as a kind of duration optimization policy, since the DRI-MSC policy based phase duration varies according to different traffic flows adaptively.

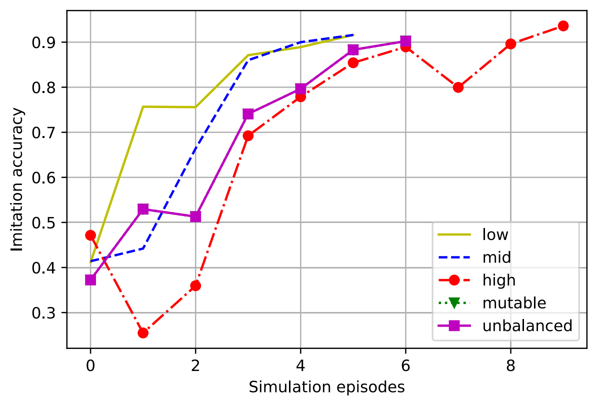
Next, we vary the number of intersections and compare the performance of the DRI-MSC model in Fig. 12.

Fig. 12 shows that the DRI-MSC is able to converge along with iterations. For the traffic network with no more than 4 intersections, the DRI-MSC model can stably converge to a point with good global efficiency within 200 simulation episodes. Since the action space gradually increases with more intersections, the convergence rate slows down with the expansion of traffic network and more training iterations are needed to explore the model. The DRI-MSC model can also converge with 6 and 9 intersections, if more simulation episodes are reached.

Furthermore, to give more insights into the DRI-MSC model, we evaluate the detailed policies and mechanisms of the DRI-MSC model respectively. Firstly, we evaluate the efficacy of the imitation learning, where different flows and various numbers of intersections are used to test the



(a) Losses of different flows.



(b) Accuracies of different flows.

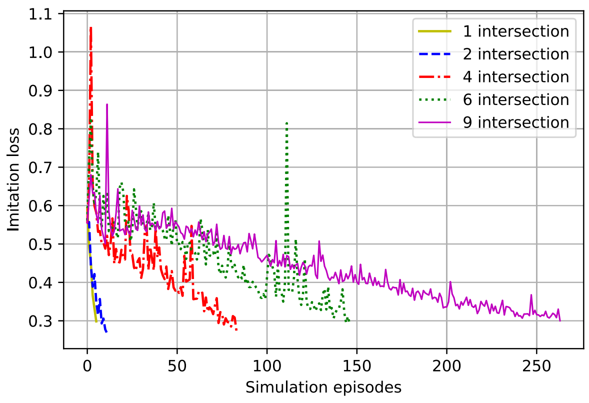
FIGURE 13. **Performance of imitation learning method of single intersection model with different traffic flows.**

accuracies brought by imitating the existing policies. Fig. 13 describes the variation of errors and accuracies of single intersection model with different traffic flows.

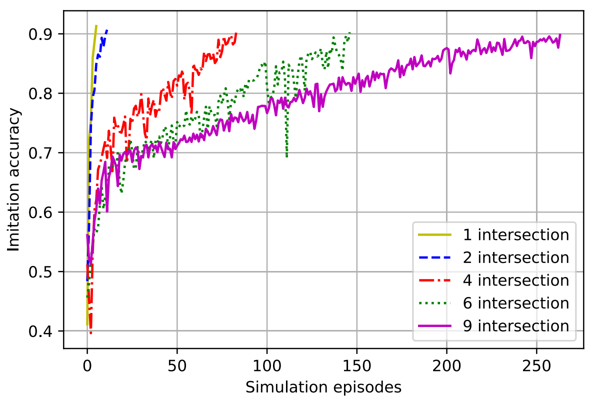
It can be seen from Fig. 13 that the imitation loss decreases and the accuracy increases with simulation episodes going on, and then they both converge to a satisfactory solution. Fig. 14a and 14b describe the performance with different numbers of intersections under the same traffic flow level.

Fig 14 shows the same tendency with Fig 13. Although the convergence speed becomes slower with more intersections, a satisfactory accuracy can still be ultimately obtained at the end. Therefore, the imitation method performs well under the situations of various intersections with different traffic flows.

As shown in Fig. 10a, for the DRI-MSC model, the left part of the vertical magenta dotted line uses imitation learning as the pre-training, and the right part uses RL for fine-tuning. We can see that the DRI-MSC model converges significantly faster than the DR-MSC model. It can also be noticed that the converging process of the DRI-MSC model is more stable with less fluctuation. For further analysis, we adopt entropy shown in Fig 15a to quantify the convergence of the probability based control policies. It is obvious that the entropy of the DRI-MSC model is distinctively lower than that of the

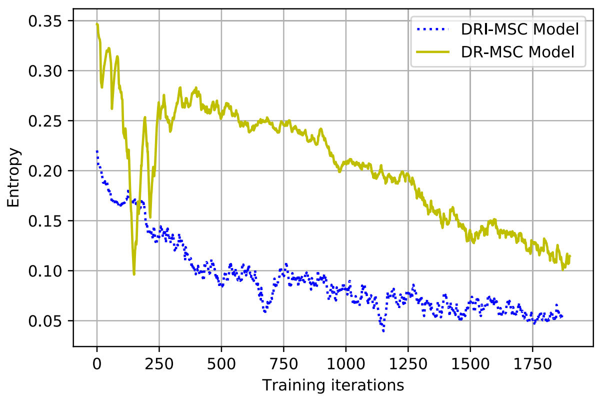


(a) Losses of different intersection numbers.

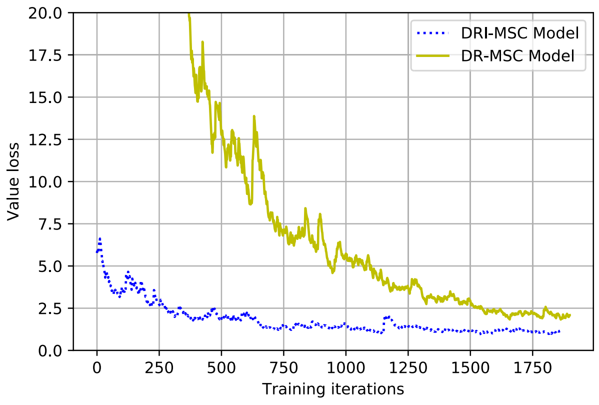


(b) Accuracies of different intersection numbers.

FIGURE 14. Performance of imitation learning method with different number of intersections with traffic flow fixed as low level.



(a) Entropy



(b) Value loss

FIGURE 15. Comparison between DRI-MSC model and RL based DR-MSC model.

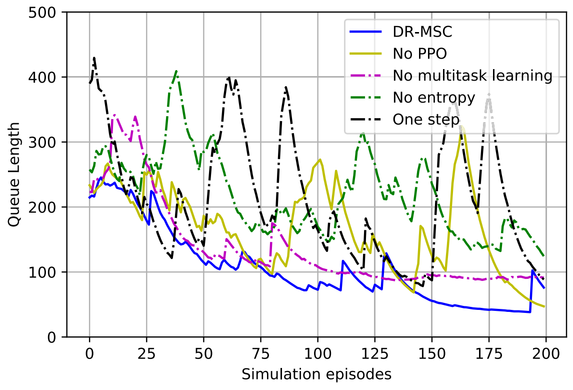


FIGURE 16. Comparison of different boosting methods.

DR-MSC model, meaning that the policy already converges to a good point via the pre-training. Therefore, imitation learning can help the model skip many redundant searches.

For the actor critic method used in the training of RL, the policies and values converge in complementary to avoid fluctuation. We also compare the convergence of value loss, which is demonstrated in Fig. 15b. Although the value loss is not taken into account in the pre-training, it can also be reduced by imitation learning. This is mainly due to the fact that the actor and critic share parameters, and the gradients of value and policy move roughly in the same direction. Therefore, both the policy and value benefit from the pre-training.

For methods of boosting the training for multi-intersection, we evaluate the effects of multi-step estimation, multi-task learning and PPO method in the overall performance of DR-MSC model respectively in Fig. 16. For the model without PPO, we use simple policy gradient instead. For the model without multi-task learning, we use different neural networks to output the actions of the four intersections and the critic value v. For the model without entropy, we add an l_2 penalty with coefficient 0.01.

In Fig. 16, the DR-MSC model works best, and PPO method can significantly improve the convergence of the model. Moreover, we can see that the DR-MSC model converges slightly faster than the model without multi-task learning method, whose overall queue length can not be lower than 80m either. This is because that multi-task learning method can learn a more general representation for all the tasks to avoid overfitting, and reduce the task-dependent noise which negatively affect the overall performance. Finally, the model with l_2 regularization converges much slower than DR-MSC model, since the soft RL method penalizes the policy directly

TABLE 3. Performance of different models.

	Flow	Low	Middle	High	Mutable	Unb
QL	Fixed-time 20	75.30	115.30	849.21	153.77	621.01
	Fixed-time 40	134.62	202.48	810.95	183.34	597.60
	Rule	36.83	102.39	442.23	105.69	103.22
	DRI-MSC	**30.90**	**86.91**	**342.23**	**101.54**	**85.36**
AWT	Fixed-time 20	18.09	18.42	52.00	21.59	36.23
	Fixed-time 40	33.43	32.80	48.58	32.91	47.39
	Rule	8.84	14.97	37.65	14.16	17.18
	DRI-MSC	**7.62**	**12.59**	**34.66**	**11.68**	**11.98**
AFC	Fixed-time 20	55.23	78.27	244.05	83.69	178.00
	Fixed-time 40	69.62	99.27	245.57	90.79	176.57
	Rule	47.01	75.47	174.81	69.79	71.28
	DRI-MSC	**45.77**	**71.82**	**152.63**	**71.08**	**66.83**

which is more effective compared with the penalty to the parameter space.

Lastly, Table 3 is given to further test the efficiency of the proposed model under different flow conditions of isolated intersection, where ''QL'', ''AWT'', ''AFC'', ''Rule'', ''IL'' and ''Unb'' represents queue length, average vehicle waiting time, average vehicle fuel consumption, rule-based policy defined in (6), and unbalanced flow respectively. Two popular fixed timing policies are also compared.

Table 3 demonstrates that the fixed timing policy performs well only under a few certain conditions. In contrast, the DRI-MSC model shows to be advantageous in various environments, especially under the condition of large flow and unbalanced flow. We can also see that the imitation learning can achieve similar results as the rule based model. The DR-MSC model performs well in all the conditions and goes beyond the rule based model, meaning that the data driven RL method can even achieve a level surpassing the human design. Generally, the DRI-MSC model performs best in all the 3 indexes given.

V. CONCLUSION

This article proposes a novel end-to-end model based on deep reinforcement learning and imitation learning for multi-intersection signal control under V2X based traffic systems. Instead of using a huge matrix, the proposed model introduces tensors to extract information as the inputs, and multi-dimensional boolean output is used to simplify the representation without information loss. In the training, imitation learning is used as the pre-training, which helps omit various redundant searches and significantly accelerate the convergence, where the PPO method is also employed to further enhance the performance. A series of numerical experiments verify the effectiveness and advantages of the proposed method with respect to both faster convergence and higher accuracy.

In future works, the incomplete information environment can be potentially considered due to the low permeability of V2X devices in real world.

REFERENCES

[1] P. Poumanyvong, S. Kaneko, and S. Dhakal, ''Impacts of urbanization on national transport and road energy use: Evidence from low, middle and high income countries,'' *Energy Policy*, vol. 46, pp. 268–277, Jul. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421512002649

[2] D. Pojani and D. Stead, ''Sustainable urban transport in the developing world: Beyond megacities,'' *Sustainability*, vol. 7, no. 6, pp. 7784–7805, Jun. 2015, doi: 10.3390/su7067784.

[3] S. Bitam and A. Mellouk, ''ITS-cloud: Cloud computing for intelligent transportation system,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 2054–2059.

[4] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, ''The 5G-enabled tactile Internet: Applications, requirements, and architecture,'' in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.

[5] C. Liu, K. T. Chau, D. Wu, and S. Gao, ''Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies,'' *Proc. IEEE*, vol. 101, no. 11, pp. 2409–2427, Nov. 2013.

[6] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, ''Vehicle-to-everything (v2x) services supported by LTE-based systems and 5G,'' *IEEE Commun. Standards Mag.*, vol. 1, no. 2, pp. 70–76, Jul. 2017.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[8] Y. Huo, J. Hu, G. Wang, and J. Chen, ''A traffic signal control method based on asynchronous reinforcement learning,'' in *Proc. CICTP*, Jul. 2018, pp. 1444–1453.

[9] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, ''Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown toronto,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[10] X. Liang, X. Du, G. Wang, and Z. Han, ''Deep reinforcement learning for traffic light control in vehicular networks,'' 2018, *arXiv:1803.11115*. [Online]. Available: http://arxiv.org/abs/1803.11115

[11] P. Mannion, J. Duggan, and E. Howley, ''An experimental review of reinforcement learning algorithms for adaptive traffic signal control,'' in *Autonomic Road Transport Support Systems*. Basel, Switzerland: Springer, 2016, pp. 47–66.

[12] J. Zeng, J. Hu, and Y. Zhang, ''Adaptive traffic signal control with deep recurrent Q-learning,'' in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1215–1220.

[13] 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.

[14] W. Genders and S. Razavi, ''Using a deep reinforcement learning agent for traffic signal control,'' 2016, *arXiv:1611.01142*. [Online]. Available: http://arxiv.org/abs/1611.01142

[15] M. Liu, J. Deng, U. Deng, X. Ming, X. Z. Northeastern, and W. Wang, ''Cooperative deep reinforcement learning for tra ic signal control,'' in *Proc. 23rd ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, Halifax, NS, Canada, 2017.

[16] E. van der Pol, ''Deep reinforcement learning for coordination in traffic light control,'' M.S. thesis, Inform. Inst., Univ. Amsterdam, Amsterdam, The Netherlands, 2016.

[17] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, ''Toward A thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control,'' in *Proc. 34th AAAI Conf. Artif. Intell., AAAI, 32nd Innov. Appl. Artif. Intell. Conf., IAAI, 10th AAAI Symp. Educ. Adv. Artif. Intell., EAAI*, New York, NY, USA, Feb. 2020, pp. 3414–3421. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/5744

[18] F.-X. Devailly, D. Larocque, and L. Charlin, ''Ig-rl: Inductive graph reinforcement learning for massive-scale traffic signal control,'' 2020, *arXiv:2003.05738*. [Online]. Available: https://arxiv.org/abs/2003.05738

[19] D. Lee, N. He, P. Kamalaruban, and V. Cevher, ''Optimization for reinforcement learning: From a single agent to cooperative agents,'' *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 123–135, May 2020.

[20] A. Costandoiu and M. Leba, ''Convergence of v2x communication systems and next generation networks,'' *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 477, Art. no. 012052, Feb. 2019, doi: 10.1088%2F1757-899x%2F477%2F1%2F012052.

[21] Y. Zhang, Q. H. Vuong, K. Song, X.-Y. Gong, and K. W. Ross, ''Efficient entropy for policy gradient with multidimensional action space,'' in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)-Workshop Track*, Jun. 2018.

[22] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, ''Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,'' *ACM Trans. Graph.*, vol. 37, no. 4, p. 143:1–143:14, Jul. 2018, doi: 10.1145/3197517.3201311.

[23] OpenAI. (2018). *Openai Five*. [Online]. Available: https://blog.openai.com/openai-five/

[24] N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, *arXiv:1703.09035*. [Online]. Available: http://arxiv.org/abs/1703.09035

[25] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. 12th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA: MIT Press, 1999, pp. 1057–1063. [Online]. Available: http://dl.acm.org/citation.cfm?id=3009657.3009806

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[27] L. C. Baird, "Reinforcement learning in continuous time: Advantage updating," in *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, Jun. 1994, pp. 2448–2453.

[28] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. AISTATS*, 2011, pp. 627–635.

[29] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *Proc. 10th Int. Conf. Mach. Learn.*, San Mateo, CA, USA: Morgan Kaufmann, 1993, pp. 41–48.

[30] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[31] A. Rapoport and A. Chammah, "Prisoner's dilemma: A study in conflict and cooperation," in *Ann Arbor Paperbacks*. Ann Arbor, MI, USA: Univ. Michigan Press, 1965. [Online]. Available: https://books.google.com/books?id=Um19AAAAMAAJ

[32] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, no. 6, pp. 503–515, 1954. [Online]. Available: https://projecteuclid.org:443/euclid.bams/1183519147

[33] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*. [Online]. Available: https://arxiv.org/abs/1506.02438

[34] G. Tesauro, "Temporal difference learning and TD-gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, Mar. 1995, doi: 10.1145/203330.203343.

[35] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proc. 19th Int. Conf. Mach. Learn.*, San Francisco, CA, USA: Morgan Kaufmann Publishers, 2002, pp. 267–274. [Online]. Available: http://dl.acm.org/citation.cfm?id=645531.656005

[36] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, F. Bach and D. Blei, Eds. Lille, France: PMLR, Jul. 2015, pp. 1889–1897. [Online]. Available: http://proceedings.mlr.press/v37/schulman15.html

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: https://arxiv.org/abs/1707.06347

[38] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 22–31. [Online]. Available: http://dl.acm.org/citation.cfm?id=3305381.3305384

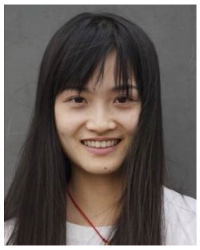
[39] B. D. Ziebart, "Modeling purposeful adaptive behavior with the principle of maximum causal entropy," Ph.D. dissertation, Dept. Mach. Learn., School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2010, Art. no. aAI3438449.

[40] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*. [Online]. Available: https://arxiv.org/abs/1812.05905

[41] 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. 33rd Int. Conf. Mach. Learn.*, vol. 48. New York, NY, USA, Jun. 2016, pp. 1928–1937.

[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

YUSEN HUO received the M.S. degree from Tsinghua University, in 2019. His research interests include intelligent transportation systems, reinforcement learning, and deep learning methods.



QINGHUA TAO received the B.S. degree from Central South University, in 2014, and the Ph.D. degree from Tsinghua University, in 2020. From 2017 to 2018, she was a Visiting Ph.D. Student with ESAT, KU Leuven, Belgium. Her research interests include modeling and training of piecewise neural networks, sparsity regularization with piecewise linearity, derivative-free optimization for black-box problem, and other relating machine learning methods and their applications.

JIANMING HU (Member, IEEE) received the B.E., M.E., and Ph.D. degrees, in 1995, 1998, and 2001, respectively. He worked with The Chinese University of Hong Kong, from 2004 to 2005, as a Research Assistant, and visited PATH, University of California at Berkeley, for one year, as a Visiting Scholar. He is currently an Associate Professor with the Department of Automation (DA), Tsinghua University. He has presided and participated over 20 research projects granted from the Ministry of Science and Technology, China, the National Natural Science Foundation of China, and other large companies with over 30 journal articles and over 90 conference papers. His recent research interests include networked traffic flow, large scale traffic information processing, intelligent vehicle infrastructure cooperation systems (V2X or connected vehicles), and urban traffic signal control. Based on his research achievements, he received two Science and Technology Improvement Awards from the Ministry of Education and the ITS Association of China, in 2016 and 2012, respectively, the First Class Award of Excellent Teaching Skill Contest in Beijing, and many awards from Tsinghua University.