IEDQN: Information Exchange DQN with a Centralized Coordinator for Traffic Signal Control

Donghan Xie
Department of Control and Systems Engineering
Nanjing University, Nanjing, China
donghanxie@smail.nju.edu.cn

Zhi Wang
Department of Control and Systems Engineering
Nanjing University, Nanjing, China
zhiwang@nju.edu.cn

Chunlin Chen
Department of Control and Systems Engineering
Nanjing University, Nanjing, China
clchen@nju.edu.cn

Daoyi Dong
School of Engineering and Information Technology
University of New South Wales, Canberra, Australia
daoyidong@gmail.com

***Abstract*—Finding the optimal control strategy for traffic signals, especially for multi-intersection traffic signals, is still a difficult task. The use of reinforcement learning (RL) algorithms to this problem is greatly limited because of the partially observable and nonstationary environment. In this paper, we study how to eliminate the above influence from the environment through communication among agents. The proposed method, called Information Exchange Deep Q-Network (IEDQN), has a learning communication protocol, which makes each local agent pay unbalanced and asymmetric attention to other agents' information. Besides the protocol, each agent has the ability to abstract local information from its own history data for interacting, which means that the communication can avoid the dependent instant information and it is robust to the potential time delay of communication. Specifically, by alleviating the effects of partial observation, experience replay can recover to good performance. We evaluate IEDQN via simulation experiments in the simulation of urban mobility (SUMO) in a traffic grid, and it outperforms the comparative multi-agent RL (MARL) methods in both efficiency and effectiveness.**

***Index Terms*—Adaptive traffic signal control (ATSC), multi-agent reinforcement learning (MARL), communication learning.**

I. Introduction

Traffic congestion is a growing problem and it causes huge economic losses and people's time wastes besides environmental pollution in urban daily life. However, there is not a good method to alleviate this problem. Adaptive traffic signal control (ATSC) is attracting wide interest in recent years, which aims to reduce the total delay time or the queue length of all travelers in the whole transportation system by optimizing the policies of traffic signals. According to the current traffic dynamic conditions, ATSC controls the timings or the orders of signal phases to ease congestion. Besides the traditional pre-timing or actuated control methods, SCOOT [1] and SCATS [2] are two of the most widely used ATSC systems in numerous cities across the world. Although these two methods could take account of real-time traffic conditions of the local intersection, they have a limited degree to describe the traffic situation and cannot deal with complex traffic dynamics.

Since the 1990s, computational intelligence methods like genetic algorithm [3], fuzzy logic [4, 5], and several other methods have been used for the ATSC problem. Nevertheless, they do not have satisfactory performance in terms of efficiency and effectiveness. Reinforcement learning (RL) [6] is considered as one of the promising methods to the ATSC problem because of its ability to deal with the sequential decision-making problem under the environment with an unknown model like a traffic system. The central idea of RL is that, under the framework of Markov decision process (MDP), an agent learns to maximize its expected return from the environment and during this learning process it optimizes the decision policy by trial and error [7–9]. Traditional RL algorithms have the strength to cope with the sequential decision problems by using tabular methods or linear approximations, while its power of decision is limited to low-dimensional problems. Contrast to the needs for extracting low-dimensional features by hand, deep neural network (DNN) [10] has the strength of its end-to-end feature representation ability. To utilize this abstraction ability to the high-dimensional input from the environment, deep reinforcement learning (DRL) [11] was proposed to combine RL and DNN to work in complex control tasks [12]. With the experience replay mechanism [13, 14], DRL could use its history experiences more efficiently.

As there are many intersections in a traffic system, the application of RL in ATSC has two main ways. One way is to see the whole system as a single centralized agent and there will be only this single agent making decisions. The agent receives the joint observation of all intersections as the input and it determines the joint action for all local intersections. This large-scale control can cause the curse of dimensionality. The other way is to use the view of the multi-agent problem to investigate the ATSC problem. While this perspective views each intersection as a single agent, the observation of each agent received from the environment is no longer a full observation of the entire system, but only a

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

partial one. It means that the framework of MDP does not suit this problem anymore, and the partially observable Markov decision process (POMDP) [15] may be better. However, there are many difficulties to directly utilize RL algorithms to a POMDP problem, such as the bad convergence policy caused by a nonstationary environment or the ineffectiveness of experience replay. Some mechanisms like coordination, communication, and incremental learning [16, 17] have been introduced to alleviate the negative effects of partial observation and nonstationary environment, which makes RL fit problems of this type.

In this paper, we propose a communication-learning-based method to a large-scale ATSC problem by alleviating the effects of partial observation, called Information Exchange Deep Q-Network (IEDQN). There are three agents, one for information extracting, another for information exchanging and the third for learning policies from the comprehensive input concatenated by local observation and the exchanged information called message. Except the centralized information exchanging agent, the other two agents are both decentralized and their parameters are shared among all local intersections. The message protocol can be learned automatically by information exchanging network so that it can achieve better performance than using raw observations from neighborhoods, not only in the effectiveness of this information but also in the communication efficiency considering the dimensions of messages. The comprehensive input of policy agents, especially the messages according to the learned protocol, could make the agent have a global observation to some degree and that makes DQN work. It is worth mentioning that all the so-called information is non-instant information from the last time step so that the possible time delay of communication has little damage to the decision-making process. Experiments demonstrate that our method can find better policy than existing methods such as IQL [18] and MA2C [19].

II. Related Work

The direct way to utilize RL algorithms in the multi-intersection ATSC problem is centralized methods which regard the entire system as a single RL agent [20]. By using the coordination graph, Kuyer et al. [21] and Van der Pol and Oliehoek [22] first trained a coordination model between two joint agents and then extended it to the centralized joint coordination over the global joint action. It faces scalability issues when the number of agents grows.

To fit large-scale problems better, independent RL methods are proposed in which each RL agent only controls a single intersection in the traffic system. Various independent Q-learning methods with different function approximations [23–25] are investigated for ATSC. Li et al. [26] combined DQN with stacked auto-encoders to control a single intersection. The recently proposed method *Intellilight* in [27] designed a phase gate and memory palace to solve the potential problem of ignorance of the current phase and unbalanced memory buffer.

Though individual and independent methods perform well in control tasks of a single intersection, the phenomenon of partial observation and nonstationary environment influences their performance in multi-agent scenarios. Communication-based methods are proposed to avoid the above influence. El-Tantawy et al. [28] proposed an indirect heuristic communication mechanism, by using an estimator model of neighbor policies. Arel et al. [29] and Wiering [30] added neighbor states directly to the local agent's state. Nishi et al. [31] used hidden states abstracted by graph convolutional neural network (GCN) of neighbors. Chu et al. [19] utilized not only neighbors' states but also their past policies, besides the extension of IQL to IA2C. However, the influence between local agents and other agents is unbalanced and asymmetric as investigated in [32], and the above communication methods pay equal attention to all the neighbors, which might cause the aliasing phenomenon that was triggered by partial observation to be solved not well.

III. Information Exchange Deep Q-Network

The difficulty to solve an ATSC problem is mainly concentrated on the partially observable and nonstationary environment. Our method investigates the communication-learning mechanism among agents to make MARL work better for this problem. The entire model can be divided into three parts from bottom to top layers, which are the decentralized Information Abstraction Blocks, a centralized Information Exchange Net and the decentralized Q-Prediction Nets, for short as ***Info-Block***, ***Mess-Net*** and ***Q-Net***, respectively. Via this communication-learning method, the problems caused by POMDP and nonstationary environment could be alleviated.

A. Infomation Abstraction

Although using neighbor agents' instant raw data from the environment is popular in communication-based MARL methods, the use of this instant information faces two main shortcomings. One is that raw data from the environment often mean less amount of information with a high dimension. The other is that the use of instant information is not robust when the communication time delay appears. For the first one, most proposed methods rely on the shallower layers of each agent to abstract the useful information, while it means that the input dimension of each agent is higher and the feature abstraction layers might ignore or not give enough attention to its own observation by contrast to the entire input's high dimension. For the second one, most methods would like to assume that the communication is ideal and the delay would not appear.

To improve the robustness of communication among agents, we propose to use the non-instant information in place of the instant information, which means that the observation and Q-values of this local agent in the last timestep are used rather than the current timestep's. In contrast to that every neighbor of the local agent needs to repeat the procedure to abstract the raw data from a local agent, we propose that each local agent has an information abstraction block to represent its raw data into a lower dimension as shown in Fig. 1. This information

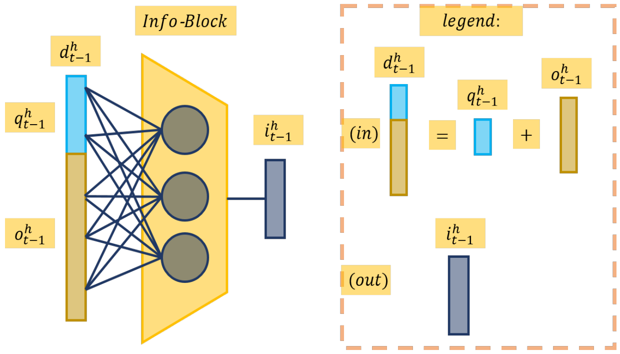


Fig. 1. The model of Info-Block.

$i_t^h = ReLU(W^h d_t^h + b^h)$ is going to be sent to all agents according to the learned communication protocol, where for each agent h its history data d_t^h is concatenated by its current observation and Q-values as $d_t^h = \{o_t^h, q_t^h\}$. By sharing this *Info-Block* among all agents as $W^1 = ... = W^h = ... = W^n$ and $b^1 = ... = b^h = ... = b^n$, the results of this block will be highly structured. The information can be formulated as

$$i_t^h = ReLU(W d_t^h + b),$$

or in an information matrix form, like

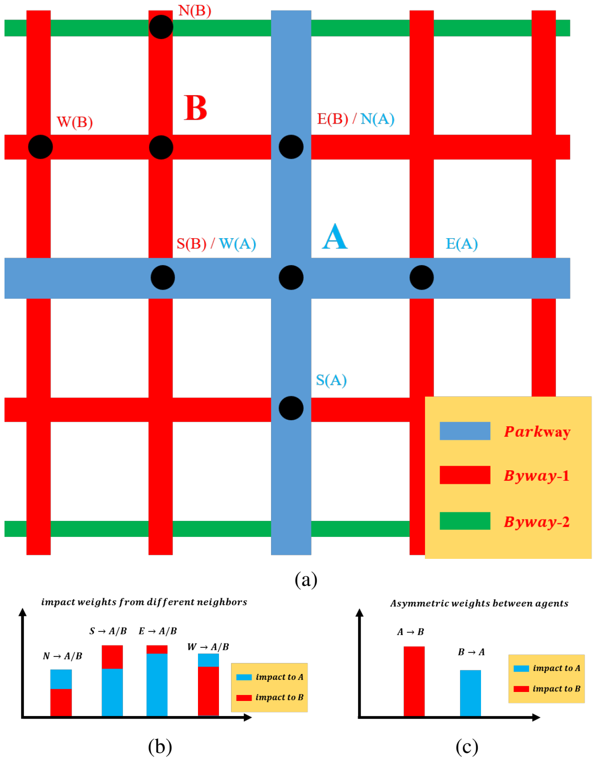
$$I_t = ReLU(W D_t + b). \tag{1}$$

To the end, as the input is combined with raw observation of the local agent and Q-values of the last timestep, this abstraction block also has the ability to describe a potential prediction of local agent's observation in the current timestep.

B. Information Exchange

Through the *Info-Block*, each agent has the ability to structurally summarize useful information in a lower dimension. However, how to use this useful information to conduct high-quality communication also needs designing. In contrast to the traditional interaction only between neighbors, we introduce a coordinator agent that learns the protocol automatically to design an efficient communication mechanism. As each intersection is in a different location in the traffic system, each intersection pays different attention to other agents in this system, which reflect not only in the weights of its neighbors but also in the number of agents it focuses on. If the information from *Info-Block* is fed directly to their neighbors, every agent will focus on different agents in an unbalanced sensitive degree as shown in Fig. 2b. As a result, it needs to design every agent according to its role in the current given traffic network environment, which means that it cannot be generalized to other occasions because the topological structure varies. Also, there is a problem of different sensitive regions, which is due to different agents showing various sensitive degrees to the same attention regions as shown in Fig. 2c.

We propose a centralized coordinator that can learn a protocol for the communication in the whole system as shown in Fig. 3. It means that, for every message receiver, the coordinator learns special information impact factors from all other agents. As the receiver differs, the impact factors differ, too. It can be formulated as



(a)

(b)

(c)

Fig. 2. The problem of unbalanced and asymmetric influence among agents in ATSC. (a) Intersections in different locations in a real-world traffic network. (b) Unbalanced influence to intersection B. (c) Asymmetric influence between intersections A and C.

$$m_t^h = f^h(I_t; \theta^h). \tag{2}$$

Though impact factors differ with the receivers, the output of this *Mess-Net*, which is called message, could still be seen as structured data in the next part of our work. It means that each receiver receives a message vector in the same formulation from the entire system, regardless of where the message vector comes from and how important a role other agent plays in this vector. Such a highly structured message is beneficial for our index-free and location-ignored method.

C. Q-Net

Like *Info-Block*, the *Q-Net* is also a parameter sharing network, which merges all differences into the centralized *Mess-Net* and leaves the rest common knowledge, regardless of the location or index of local agents, to *Info-Block* and *Q-Net*. There is only an IQL agent with shared parameters working for all intersections, which is shown in Fig. 4. We suppose that the integration of messages and current local observation could be seen as a fully observable state which alleviates the partially observable problem. While we suppose the messages from *Mess-Net* imply a potential prediction of current policies, the environment is no longer a nonstationary one, which means

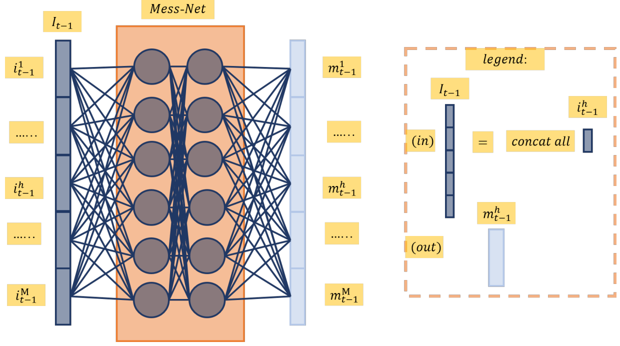


Fig. 3. The model of Mess-Net.

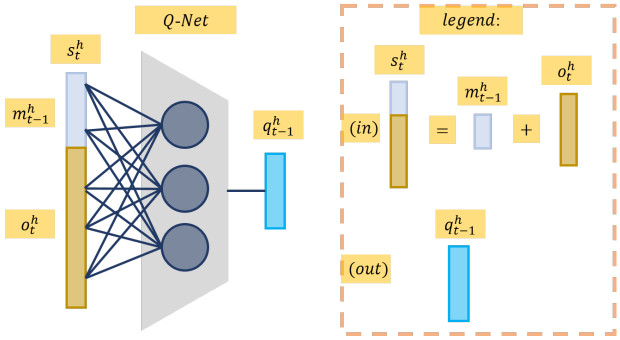


Fig. 4. The model of Q-Net.

experience replay could play a role in training the IQL agent. The output of this *Q-Net* is the prediction of Q-values for each action under the current integrated state, which can be formulated as

$$q^h_t = \left\{q^h_t\left(s^h_t, a^h; \sigma\right)\right\}, \forall a^h \in \mathbb{A}^h;\ s^h_t = \{o^h_t, m^h_{t-1}\}, \quad (3)$$

where the superscript h means the value of this variety differs for each different local agent h. In RL, the Q-value function is iteratively updated by using *Bellman equation* as

$$Q_{i+1}(s,a) = \mathbb{E}\left[r + \gamma \max_{a'} Q_i(s', a'|s,a)\right]. \quad (4)$$

The target Q-value can be formulated as

$$y^h_t = \begin{cases} r^h_t, & \text{if } s_{t+1} = s_T, \\ r^h_t + \gamma \max_{a^h_{t+1}} q^h_{t+1}, & \text{otherwise,} \end{cases} \quad (5)$$

and the parameters W, b, θ, σ are all trainable during minimizing the loss function:

$$L^h_t(\sigma, \theta^h, W, b) = \mathbb{E}\left[(y^h_t - q^h_t(o^h_t, m^h_{t-1}, a^h_t; \sigma))^2\right]. \quad (6)$$

Our method combined with these three above parts can be described in the form of Algorithm 1, and the entire model is shown in Fig. 5. Note that each upper letter means a concatenation of corresponding lower letters as

$$I_t = \{i^1_t, ..., i^M_t\},$$

$$D_t = \{d^1_t, ..., d^M_t\}, \text{ where } d^h_t = \{q^h_t, o^h_t\},$$

and Q_t, O_t, M_t, A_t, R_t are also the same form.

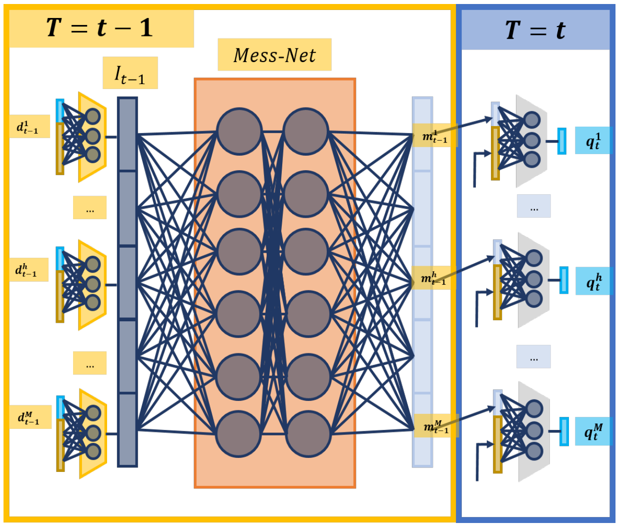


Fig. 5. The entire model of proposed IEDQN. Specifically, all the *Info-Block* and *Q-Net* are sharing parameters respectively.

D. Communication Cost Analysis

We propose a communication-learning-based MARL method for ATSC, and this communication mechanism is based on non-instant data from the last timestep. Via the centralized coordinator as *Mess-Net*, two types of communication channels are established. In one channel *Info-Block* plays the role of the sender and *Mess-Net* plays the role of the receiver. In the other one, *Mess-Net* plays the role of the sender and *Q-Net* plays the role of the receiver. Suppose the number of agents in the entire traffic system is *M*, the frequency of communication during a single decision process is $2M$. Popular methods consider the communication between neighbors, which is at least $4(M - \sqrt{M})$ in a square traffic grid. The ratio of communication frequency between our methods and these neighbors-based communication methods is

$$r = \frac{M}{2(M - \sqrt{M})} \rightarrow \frac{1}{2},$$

which indicates that our method can reduce about 50% communication cost as the number of agents in the system grows.

IV. Experiments

We conduct simulation experiments in a 5x5 traffic grid net based on an open source traffic simulation application called Simulation of Urban MObility (SUMO) [33], which is widely used in several research topics like route choice, traffic light algorithm and so on.

To apply RL algorithms to ATSC, the first to be considered is a good definition of RL agent, which we call RL settings, about state space, action space, and reward. Besides the RL settings, the traffic network and traffic data also need to be considered.

A. RL Settings

About RL settings, we use the similar definition in [19].

Algorithm 1: Information Exchange DQN

```
Initialize replay memory buffer B
Initialize *Info-Block*, *Mess-Net* and *Q-Net* with random weights $W, b, \theta, \sigma$
for episodes = $1, ..., N$ do
    Initialize $D_0$ and $O_1$
    Initialize $I_0$ using $D_0$
    Initialize message vectors $m_0^h = f^h(I_0; \theta^h), \forall h \in M$ and send it to agent $h$
    for $t = 1, ..., T$ do
        Calculate $Q_t$ according to $M_{t-1}$ and $O_t$
        for each agent $h = 1, ..., M$ do
            With probability $\epsilon$ select a random action $a_t^h$ otherwise select action $a_t^h = \arg\max_{a_t^h} q_t^h$
        end
        Execute actions $A_t$, obtain rewards $R_t$ and observations $O_{t+1}$
        Store transition $(D_{t-1}, O_t, A_t, Q_t, O_{t+1})$ in B
        Sample random minibatch of transitions $(D_{j-1}, O_j, A_j, Q_j, O_{j+1})$ from B
        for each agent $h = 1, ..., M$ do
            Calculate message vector $m_j^h$ using (2)
            Calculate target Q-value with respect to (5)
            Calculate loss $L_t^h(\sigma, \theta^h, W, b)$ using (6)
            Update parameters $\sigma, \theta^h, W, b$ according to $L_t^h(\sigma, \theta^h, W, b)$
        end
        Calculate $I_t$ and $M_t$
        Send message vector $m_t^h$ to agent $h$
    end
end
```

1) State Space: The local observation, after combining the ideas of [20] and [34], is defined as

$$o_t^h = \{wait_t[l], wave_t[l]\}_{ih \in \varepsilon, l \in L_{ih}}, \tag{7}$$

where l is each incoming lane of intersection i, $wait[s]$ measures the cumulative delay of the first vehicle and $wave[veh]$ measures the total number of approaching vehicles along each incoming lane, within 50m to the intersection.

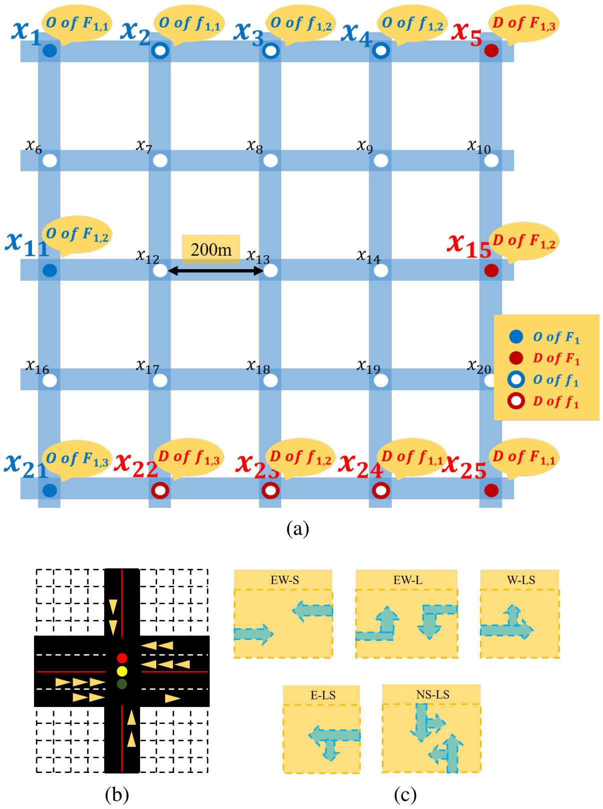
The input of *Info-Block*, which called history data d, is defined by Q-values and local observation as

$$d_t^h = \{q_t^h, o_t^h\}, \tag{8}$$

and the state of *Q-Net* is formed by local observation and message that is generated by the above history data as

$$s_t^h = \{o_t^h, m_{t-1}^h\}, \text{ where } m_{t-1}^h = f^h(I_{t-1}; \theta^h). \tag{9}$$

2) Action Space: Contrast to traditional methods' cyclic phases with only two actions as switch and not-switch, our RL action uses a one-hot vector with the same dimension of possible phases of the local intersection to determine the next phase of the local agent. If the continuous two decisions are different, an all-yellow phase appears lasting for $t_y[s]$ before the second decision's execution.



(a)

(b)

(c)

Fig. 6. The simulation environment of the experiment. (a) A traffic gird of 25 intersections with 4 example flows (two omitted flows in this pic are swapped O-D pairs as the shownw two). (b) One of the intersections. (c) Possible actions.

3) Reward Definition: The definition of reward can be formulated as follows:

$$r_t^h = -\sum_{ih \in \varepsilon, l \in L_{ih}} \left(queue_{t+\triangle t}[l] + a \cdot wait_{t+\triangle t}[l]\right), \tag{10}$$

where a is a trade-off coefficient, and *queue* is the measured queue length along each incoming lane. In our experiments, the coefficient a is set as $0.2[veh/s]$.

4) Other Settings: Besides the definition of state space, action space and reward, the timestep between two decisions also needs to be set. We prefer $\triangle t = 5s$ which means the interaction between RL agents and traffic environment appears every $5s$. If the next decision is different from the last one, an all-yellow phase would appear with the duration of $t_y = 2s$ to ensure a safe switch between phases.

B. Traffic Settings

As shown in Fig. 6a, the traffic network is a 5×5 grid and each intersection in this grid is uniform. The intersection is made up of E-W two-lane arterial streets and N-S one-lane avenues, with speed limits of 20m/s and 11m/s, respectively. On each intersection, there are five possible phases available, which are EW-S, EW-L, W-LS, E-LS, and NS-LS as shown in Fig. 6b.

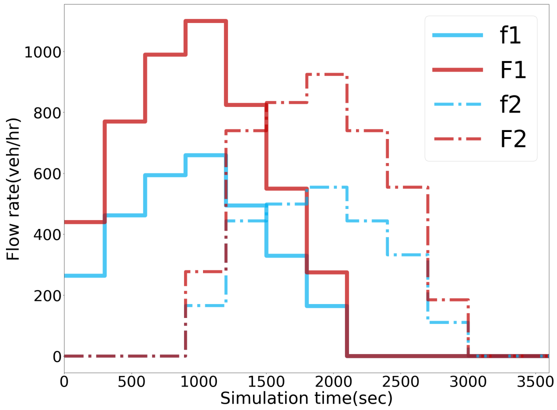


Fig. 7. Traffic flow.

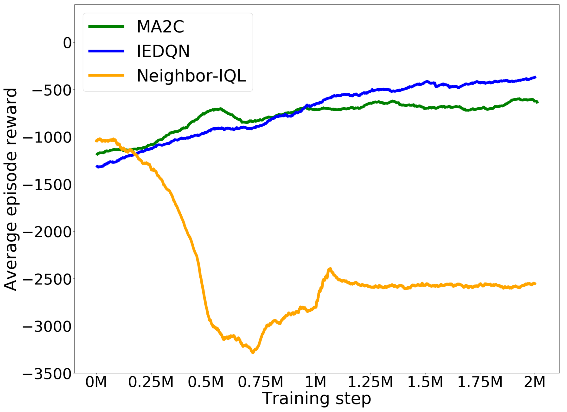


Fig. 8. Training curves of MARL methods in experiment.

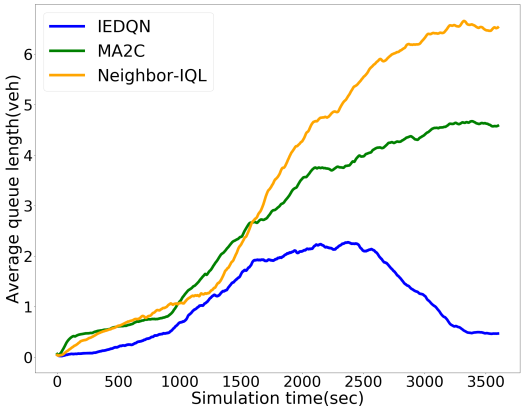


Fig. 9. Average queue length in evaluation.

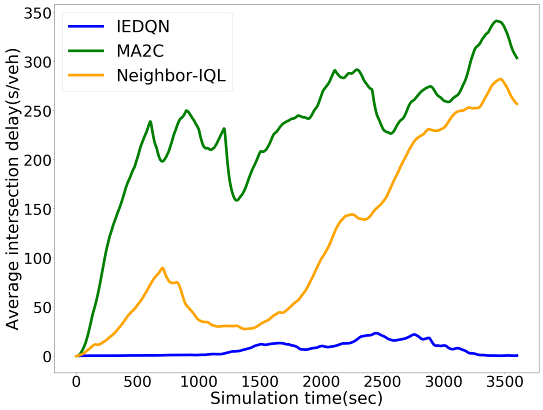


Fig. 10. Average intersection delay in evaluation.

In the entire system, the synthetic traffic flow data are combined with four time-variant traffic flow groups as shown in Fig. 7. To describe these traffic flows, we use the origin-destination (O-D) pairs description. There are three major flows F_1 with O-D pairs x_{21}-x_5, x_{11}-x_{15} and x_1-x_{25}, and three minor flows f_1 with O-D pairs x_2-x_{24}, x_3-x_{23} and x_4-x_{22}. Their opposite flows with swapped O-D pairs appear at 15min.

C. Experiment Results

In the experiment, we compare our method IEDQN with some other methods in large-scale ATSC problem. MA2C is a modified version of IA2C, which is inspired by IQL. IQL is a benchmark method in MARL, and we use this method appended with neighbors' observation [29] with DNN as an approximation function to fit Q-function, which called Neighbor-IQL. We train these methods over 2M steps, and each episode in the traffic environment is $T = 720$ steps with 3600s as $\triangle t = 5s$. Then, these trained models perform over 10 episodes to show their performance of the learned policy. The MDP settings in the experiment are $\gamma = 0.99$ and $\alpha = 0.75$. The setting of MA2C is the same as in [19]. For DQN based methods, the replay buffer size is 1000 and the sample size is $|B| = 20$, besides ϵ-greedy with linearly decaying ϵ from 1.0 to 0.01 for the first 1M steps.

1) Training Results: We show the curves of the average reward per step in each episode of the mentioned methods in Fig. 8. It can be seen that a bad convergence trend occurs in Neighbor IQL, which might be caused by a phenomenon of state aliasing as a result of the partial observation. With steps increasing, the performance of IEDQN is better than MA2C, though research shows that A2C performs better than DQN, and that means the communication mechanism of IEDQN can alleviate the influence of partial observation better than the limited communication between neighbors.

2) Evaluation Results: ATSC aims to reduce congestion. It also can be considered to decrease the average queue length and reduce the average intersection delay.

Fig. 9 shows the average queue length per simulation

TABLE I
PERFORMANCE IN EVALUATION FOR DIFFERENT METHODS

Metrics	Temporal Average			Temporal Peaks		
	IEDQN	***MA2C***	***Neighbor IQL***	***IEDQN***	***MA2C***	***Neighbor IQL***
avg. queue length [veh]	**1.15**	2.71	3.46	**2.34**	4.70	6.70
avg. intersection delay [s/veh]	**7.33**	233.80	118.35	**24.34**	347.02	284.99
avg. vehicle speed [m/s]	**4.63**	1.49	2.21	**19.22**	14.23	12.09

second over 10 evaluation episodes. When traffic demand is increasing, all of the queue lengths would grow in these three methods. Neighbor IQL obtains a bad result as the queue length is out of control. Though MA2C has no ability to solve congestion, it shows the capacity of controlling the congestion situation in an acceptable range and the situation no longer becomes worse. IEDQN performs a stable policy to control congestion in a low level and can fast alleviate the congestion.

Fig. 10 plots the average intersection delay across the entire system over evaluation episodes. It shows a positive correlation between the delay and the queue length, respectively in the three methods, which means the learned policy of each method is not only to decrease the average queue length but also the delay. What is surprising is that the performance of delay in MA2C is worse than in Neighbor IQL, which might indicate that the convergence policy of MA2C pays more attention to the queue length rather than the delay. The line of IEDQN shows its good control to delay even at peak time. We summarize the performance metrics of each method in Table I, including average queue length, average intersection delay, and average vehicle speed.

V. CONCLUSION

In this paper, we propose a communication-learning-based method, IEDQN, to solve the ATSC problem. The novel features proposed are, 1) to alleviate the influence of partially observable environment, we use a centralized coordinator with learned protocol, instead of information and protocol by hand; 2) the history data rather than instant information are used in the communication among agents, which is robust to communication time delay; 3) experience replay can work because of the alleviation of problems caused by nonstationary environment. Experiments in a synthetic traffic grid demonstrate that even the model of the ATSC problem is POMDP, the proposed IEDQN still obtains stable and optimal performance by eliminating the impact of partial observation to some degree.

REFERENCES

[1] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, "The SCOOT on-line traffic signal optimisation technique," *Traffic Engineering & Control*, vol. 23, no. 4, 1982.

[2] J. Luk, "Two traffic-responsive area traffic control methods: SCAT and SCOOT," *Traffic Engineering & Control*, vol. 25, no. 1, 1984.

[3] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers routing," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 329–342, 2004.

[4] M. B. Trabia, M. S. Kaseko, and M. Ande, "A two-stage fuzzy logic controller for traffic signals," *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 6, pp. 353–367, 1999.

[5] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 714–727, 2010.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 2nd ed. MIT press, Cambridge, MA, 2018.

[7] Z. Wang, H.-X. Li, and C. Chen, "Reinforcement learning-based optimal sensor placement for spatiotemporal modeling," *IEEE Transactions on Cybernetics*, 2019, Preprint online, Doi: 10.1109/TCYB.2019.2901897.

[8] J.-A. Li, D. Dong, Z. Wei, Y. Liu, Y. Pan, F. Nori, and X. Zhang, "Quantum reinforcement learning during human decision-making," *Nature Human Behaviour*, 2020.

[9] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1207–1220, 2008.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[12] Z. Wang, H.-X. Li, and C. Chen, "Incremental reinforcement learning in continuous spaces via policy relaxation and importance weighting," *IEEE Transactions on Neural Networks and Learning Systems*, 2019, Preprint online, Doi: 10.1109/TNNLS.2019.2927320.

[13] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[14] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2216–2226, 2018.

[15] M. T. Spaan, "Partially observable Markov decision processes," in *Reinforcement Learning*. Springer, 2012, pp. 387–414.

[16] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "Incremental reinforcement learning with prioritized sweeping for dynamic environments," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 621–632, 2019.

[17] L. Zhou, P. Yang, C. Chen, and Y. Gao, "Multiagent reinforcement learning with sparse interactions by negotiation and knowledge transfer," *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1238–1250, 2016.

[18] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," *PLoS ONE*, vol. 12, no. 4, 2017.

[19] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, 2019, Preprint online, Doi: 10.1109/TITS.2019.2901791.

[20] L. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 412–421, 2010.

[21] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2008, pp. 656–671.

[22] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," *NIPS'16 Workshop of Learning, Inference and Control of Multi-Agent Systems*, 2016.

[23] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[24] S. El-Tantawy and B. Abdulhai, "An agent-based learning towards decentralized and coordinated traffic signal control," in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, 2010, pp. 665–670.

[25] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *arXiv preprint arXiv:1611.01142*, 2016.

[26] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[27] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2496–2505.

[28] 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 Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[29] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.

[30] M. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proceedings of International Conference on Machine Learning*, 2000, pp. 1151–1158.

[31] T. Nishi, K. Otaki, K. Hayakawa, and T. Yoshimura, "Traffic signal control based on reinforcement learning with graph convolutional neural nets," in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 877–883.

[32] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," *arXiv preprint arXiv:1905.05717*, 2019.

[33] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: an overview," in *Proceedings of International Conference on Advances in System Simulation*, 2011.

[34] 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," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 732–752, 2017.