STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control

Yanan Wang, Tong Xu, *Member, IEEE*, Xin Niu, Chang Tan,
Enhong Chen, *Senior Member, IEEE*, and Hui Xiong, *Fellow, IEEE*

Abstract—The development of intelligent traffic light control systems is essential for smart transportation management. While some efforts have been made to optimize the use of individual traffic lights in an isolated way, related studies have largely ignored the fact that the use of multi-intersection traffic lights is spatially influenced, as well as the temporal dependency of historical traffic status for current traffic light control. To that end, in this article, we propose a novel Spatio-Temporal Multi-Agent Reinforcement Learning (STMARL) framework for effectively capturing the spatio-temporal dependency of multiple related traffic lights and control these traffic lights in a coordinating way. Specifically, we first construct the traffic light adjacency graph based on the spatial structure among traffic lights. Then, historical traffic records will be integrated with current traffic status via Recurrent Neural Network structure. Moreover, based on the temporally-dependent traffic information, we design a Graph Neural Network based model to represent relationships among multiple traffic lights, and the decision for each traffic light will be made in a distributed way by the deep Q-learning method. Finally, the experimental results on both synthetic and real-world data have demonstrated the effectiveness of our STMARL framework, which also provides an insightful understanding of the influence mechanism among multi-intersection traffic lights.

Index Terms—Traffic light control, mobile data mining, multi-agent reinforcement learning, graph neural network

1 Introduction

Recent years have witnessed a sharp increase in traffic congestion in most cities, which results in several negative effects like air pollution and economic losses. For instance, traffic congestion has caused the financial cost of $305 billion in 2017 in the US, $10 billion more than 2016 [1]. Along this line, optimal control of traffic lights has been widely used for reducing congestion in mobile environments [2], [3], [4], [5], [6], [7]. Traditionally, the control plan of traffic lights was predefined as fixed based on historical traffic data [8], [9], or artificially regulated by officers based on current traffic status [10]. However, these solutions might be rigid and shortsighted, or even lead to the heavy burden of manpower. Thus, a more intelligent plan is still urgently required.

Thanks to the development of data analytic techniques, nowadays, traffic light control has been supported by advanced methods like *reinforcement learning* [11], [12], [13], which effectively model the traffic status to make sequential decisions. However, though prior arts performed well, most of them are restricted to the isolated intersections without coordination. Indeed, in a real-world situation, control of traffic light will definitely impact the traffic status and then results in a chain reaction on adjacent intersections. Obviously, mutual influence among multiple intersections should not be neglected during the modeling. To that end, solutions based on *multi-agent* reinforcement learning have been designed [14], [15], which further improved the performance. However, they may still face some challenges. First, the dimensionality of action space grows exponentially with the increasing number of agents, which causes dramatic complexity. Second, though distributed models may alleviate the problem of dimension explosion, it is still difficult to formulate the coordination among multiple traffic lights.

Intuitively, when describing the correlation among multiple intersections, we realize that they could be approximately formulated as *graph* structure based on spatial adjacency on the road network as shown in Fig. 1. The graph structure can be greatly different due to different types of road connections between traffic lights. Similar to information flow in the graph, traffic volume in the current intersection can be naturally split for adjacent intersections, which results in the *spatial influence* among multiple traffic lights. Thus, when jointly controlling multiple traffic lights to optimize large scale traffic situation, it is critical to model

- *Yanan Wang is with the School of Computer Science, University of Science and Technology of China, Hefei 230027, China, and also with the Department of Management Information Systems, Eller College of Management, the University of Arizona, Tucson, AZ 85721 USA. E-mail: ynwwang@mail.ustc.edu.cn.*
- *Tong Xu and Enhong Chen are with the Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science, University of Science and Technology of China, Hefei 230027, China. E-mail: {tongxu, chenehg}@ustc.edu.cn.*
- *Xin Niu and Chang Tan are with iFLYTEK Research, IFLYTEK Company, Ltd, Hefei, Anhui 230088, China. E-mail: {xinniu2, changtan2}@iflytek.com.*
- *Hui Xiong is with the Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ 07102 USA. E-mail: hxiong@rutgers.edu.*

Manuscript received 12 Feb. 2020; revised 22 Sept. 2020; accepted 12 Oct. 2020. Date of publication 26 Oct. 2020; date of current version 5 May 2022.
(Corresponding authors: Tong Xu and Hui Xiong.)
Digital Object Identifier no. 10.1109/TMC.2020.3033782

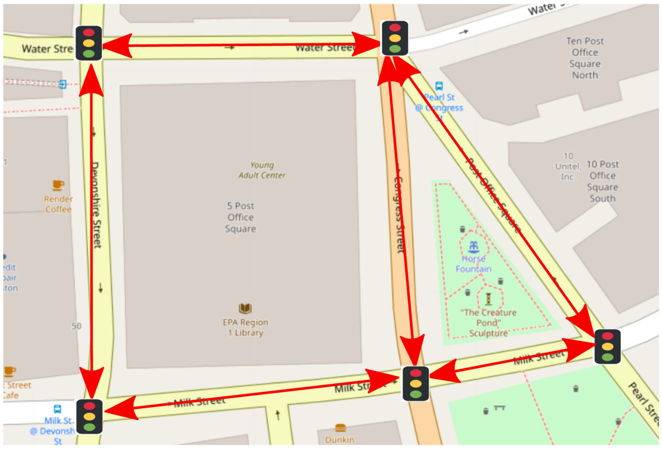


Fig. 1. Example of the spatial structure among multiple traffic lights along the road netowrk.

the cooperation structure among multiple traffic lights. Moreover, short periods are spent on traffic flow when moving to adjacent intersections, which further results in the *temporal dependency* among multiple traffic lights. Therefore, our challenge has been transferred as modeling the spatio-temporal influence among multiple intersections for intelligent traffic light control.

To that end, in this work, we propose a Spatio-Temporal Multi-agent Reinforcement Learning (STMARL) framework for multi-intersection traffic light control. Specifically, we first construct a directional traffic light adjacency graph based on the spatial structure among traffic lights. Then, historical traffic records will be incorporated with current traffic status via Recurrent Neural Network structure. Afterwards, based on the traffic information with temporal dependency, we design a Graph Neural Network based module to model the cooperation structure among multiple traffic lights, which allows the efficient relation reasoning among the traffic lights. Finally, the distributed decision for each traffic light will be made by a deep Q-learning method. Both quantitative and qualitative experiment results have demonstrated the effectiveness of our STMARL framework, which provides an insightful understanding of the influence mechanism among multi-intersection traffic lights. The technical contribution of this paper could be summarized as follows:

- To the best of our knowledge, we are among the first ones who study the spatio-temporal dependency among multiple traffic lights based on the constructed directional traffic light adjacency graph, with leveraging graph structure for better modeling cooperation mechanism between traffic lights.
- A novel multi-agent reinforcement learning framework is proposed, in which graph neural network with attention mechanism [16] for iterative relational reasoning and the recurrent neural network is incorporated to model the spatio-temporal dependency.
- Experiments on both synthetic and real-world datasets validated the effectiveness of our solution compared with several state-of-the-art methods, and further revealed some cooperation mechanism among traffic light agents.

2 Related Works

In this section, we briefly review the related works in traffic light control, methodologies of multi-agent reinforcement learning and graph neural networks.

Traffic Light Control. In the literature, traffic light control methods can be mainly divided into three types: predefined fixed-time control [8], actuated control [10] and adaptive traffic control [12], [17], [18], [19]. The predefined fixed-time control is determined offline using historical traffic data and actuated control is based on current traffic state using predefined rules to decide when to set the green phases (e.g., extending the time of green phase or setting to red phase). The main drawback of these two methods is that they do not take into account the long term traffic situation. Therefore, researchers began exploring adaptive traffic light control methods. Following this line, reinforcement learning methods have been used for traffic light control [11], [15], [17], [20], [21], [22], [23] so that the control strategy can be adaptively created based on the current traffic state.

Although reinforcement learning methods have achieved success for traffic light control in one intersection, it's still challenging for the multi-intersection traffic light control task: the curse of dimensionality in the centralized model and coordination problem in a distributed model. Kuyer *et al.* [15] and van der Pol*et al.* [24] formulated the explicit coordination among agents using max-plus [25] algorithm which estimates the optimal joint action by sending locally optimized messages among connected agents, which needed to handle the combinatorially large joint action space and was computationally expensive. In spite of using max-plus algorithm, Baker*et al.* [26] handled the partial observability of the traffic state by estimating a belief state using Bayes'rule. However, it did not model the high-order interactions between traffic lights and was computationally expensive. Khamis*et al.* [18] extended the framework of [27] using Bayesian theory and optimize traffic signal control in a multi-objective setting. However, it did not explicitly model the cooperation structure among traffic lights. Chu*et al.* [28] proposed to utilize independent advantage actor-critic (A2C) instead of Q learning for traffic light control. Although they augmented each agent's state representation with the state of its neighbors and using a spatial discount factor to adjust the global reward for each agent, they only incorporated the first-order neighbor information for each agent. Hua *et al.* [29] suggested to use max pressure [30] as a reward for traffic light control in the arterial network. Neighbor RL [22] directly concatenated the neighboring intersections' observation into their state representation. However, it did not discriminate neighbors with different traffic situations and only considered the nearest intersection. Nish *et al.* [31] proposed to used a graph convolution network to extract traffic features of distant roads. Its graph was constructed on lane level where each lane was regarded as a node and vehicle traffic movement connected to two lanes was represented as an edge. The graph size is increasingly larger as the number of intersections becomes larger. It also did not distinguish the traffic flow from different neighbors. Recently, Hua *et al.* [32] introduced a graph attention network for network-level traffic light control. However, neighbors of each agent were determined using rules with a predefined and fixed amount. The traffic flow direction on the graph was also not incorporated.

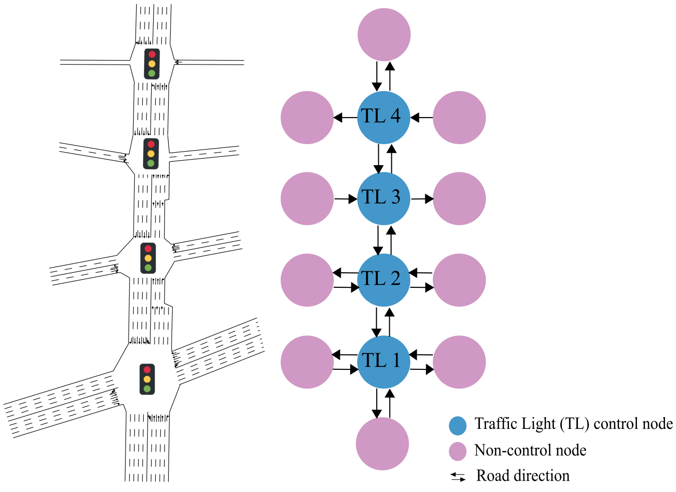
The above methods control traffic lights based on the physical infrastructure of traffic lights. Instead of using the physical infrastructure of traffic lights, some researchers [33], [34] proposed to control traffic lights with the concept of Virtual Traffic Light (VTL), which is an infrastructure-less traffic control system solely based on Vehicle-to-Vehicle (V2V) communication. In the concept of VTL, a vehicle in the intersection is selected as a leader, which is responsible for creating and controlling VTL as well as broadcasting traffic signal messages.

Different from prior arts, in this paper, we proposed STMARL to collectively learn the spatio-temporal dependency among multiple traffic lights based on a constructed intersection-level directional traffic light adjacency graph. Although we focused on controlling physical traffic lights in this paper, we can also apply the proposed method on the virtual traffic lights with a slight modification. Specifically, after selecting the leader in each intersection, we can apply the proposed method on the level of these leaders by regarding these leaders as traffic lights.

Multi-agent Reinforcement Learning. In the setting of multi-agent reinforcement learning [14], [35], [36], [37], [38], agents are optimized to learn cooperative or competitive goals. Cooperative operation is important in multi-agent systems, such as localization of networked agents and multi-object tracking [39], [40], [41], robot navigation and autonomous driving [42], [43] Independent Deep Q-Networks (DQN) [44], [45] extends DQN to multi-agent settings where each agent learns its policy independently. Although there is a non-stationary problem for independent DQN, it often works well in practice [45], [46]. To address the issue of reinforcement learning method for multi-agent settings, Lowe *et al.* [47] proposed multi-agent actor-critic for mixed cooperative-competitive environments. They adopt the framework for centralized training with decentralized execution for cooperation.

Note that previous works [38], [48] mainly design heuristic rules to decide who or how many agents the target agent communicates to, while in this paper, we learn to communicate via the existing spatial structure among agents as well as temporal dependency for multi-intersection traffic light control.

Graph Neural Networks. Our proposed method is also related to recent advances of Graph Neural Network (GNN) [49], [50], [51], [52]. GNN has been proposed to learn the structured relationship, which allows for iterative relational reasoning [53] through message passing [54] on the graph. Battaglia *et al.* [50] introduced a general framework of *Graph Networks* which unified various proposed graph network architectures to support relational reasoning and combinatorial generalization. Recently, some works are trying to explore relational inductive biases in deep reinforcement learning. Wang*et al.* [55] proposed NerveNet for robot locomotion, where it modeled the skeleton of a robot using a discrete graph structure and output actions for different nodes of this robot. Zambaldi *et al.* [56] proposed to use relational inductive biases in deep reinforcement learning agents for StarCraft II game. However, there is no explicit graph construction that is learned from the raw visual input. Comparatively, in this paper, we explicitly construct the directional traffic light adjacency graph for modeling the geographical structure information to facilitate the coordination among multi-intersection traffic light control.



(a) Spatial structure (b) Constructed graph

Fig. 2. An illustration of the traffic light adjacency graph structure in the real world.

3 Problem Statement

In this section, we will first introduce the construction of the directional *traffic light adjacency graph*, and then formally define our problem.

We first attempt to describe the road network structure in the graph perspective. In a real-world scenario, the structure of the intersections could be complicated. For instance, as shown in Fig. 2a, those roads, which are linked to the same intersection, may hold a different number of lanes (e.g., ranged from 1 to 5) and different passing constraint (two-way or one way). To describe the complicated settings, we construct the *traffic light adjacency graph* as $G = (V, E)$, as shown in Fig. 2b. Specifically, $V = \{v_i\}_{i=1}^{|V|}$ denotes a set of nodes, where v_i is the ith node's observation information. The type of nodes includes the *control nodes* which contain the traffic lights (blue nodes), and *non-control nodes* which indicate the endpoints (pink nodes). The notation of non-control nodes (end-points) is included for the representation of graphical integrity. At the same time, $E = \{(e_k, rec_k, send_k)\}_{k=1}^{|E|}$ denotes a set of edges, in which e_k indicates the kth edge's observation information, rec_k is the index of the receiver node and $send_k$ is the index of the sender node of the k-the edge. The kth edge is a directed road connecting two nodes with the type $c(k)$, meaning the number of lanes l_k in this directed road. Definitely, we use a unidirectional edge in G to present each one-way road, and each two-way road is presented by two edges with opposite directions.

Based on the constructed *traffic light adjacency graph*, we then turn to study the problem of multi-intersection traffic light control in the view of *multi-agent reinforcement learning*. Specifically, we treat each traffic light as one *agent*, and the group of traffic light agents is learned cooperatively to maximize the global reward (e.g., minimize the overall queue length in this area). Along this line, the multi-intersection traffic light control problem could be defined as a *Markov Decision Process* (MDP) for N agents within finite steps.

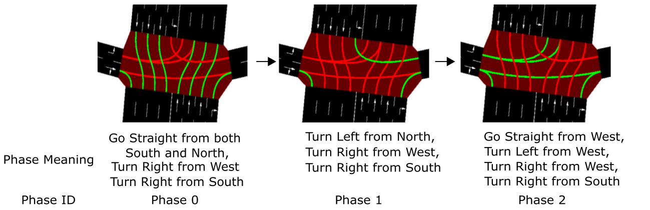

Phase Meaning
Go Straight from both South and North, Turn Right from West Turn Right from South
Turn Left from North, Turn Right from West, Turn Right from South
Go Straight from West, Turn Left from West, Turn Right from West, Turn Right from South
Phase ID
Phase 0
Phase 1
Phase 2

Fig. 3. One sample of fixed phase order in a cycle used in the real world with the meaning of each traffic light phase.

Moreover, considering that each traffic light agent receives local noisy observations in the real world [26], we further extend the MDP problem as *Partially-Observable Markov Decision Process* (POMDP), which can be defined as a tuple $(N, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{U}, \mathcal{R}, \gamma)$, in which N denotes the number of agents, and the rest are listed as follows:

- *State Space* $\mathcal{S}$: $s_{i,t} \in \mathcal{S}$ is the true system state, which is usually not observable. For agent i, the true system state consists of all the complete and accurate information in the traffic light adjacency graph (i.e., the accurate traffic information in the whole area) at time t, which is not directly accessible. Instead, the agent receives the observations.
- *Observation Space* $\mathcal{O}$: $o_{i,t} \in \mathcal{O}$ is the received observation for agent i in partially observable environment at time t.In the graph G, the observed traffic information on each edge and each node is denoted as e_k and v_i: $e_k = \{\{q_l\}_{l=1}^{l_k}, \{n_l\}_{l=1}^{l_k}, \{speed_l\}_{l=1}^{l_k}\}$, $v_i = phaseID_i$ where $q_l, n_l, speed_l$ are the queue length, number of vehicles and average speed of vehicles in lane l respectively and $phaseID_i$ is the current phase ID for node i. Phase ID is the index of a phase in the predefined phase set, where the phase is defined as a combination of valid movements for vehicles. One example of the meaning of phase is illustrated in Fig. 3. Note that the partially observability $o_{i,t}$ for agent i is defined as the observed edges information $\{e_k\}_{rec_k=i}$ whose receiver node is node i and the observed node information v_i, which represents a local view of each traffic light agent and further motivates the learning of structure dependency among traffic lights. Besides, the observed traffic information is noisy in the realistic environment, e.g., due to the noisy sensors. Therefore, the partial observability in our problem refers to the local noisy observed information for each traffic light agent.It is also straightforward to incorporate other complex features into the observation representation, while we focus on designing a novel framework for multi-intersection traffic light control in this paper.
- *Action* $\mathcal{A}$: $a_t = \{a_{i,t}\}_{i=1}^N \in \mathcal{A}$ is the joint action for all the traffic light agents at time t. There are mainly two kinds of action settings. One is to determine whether current phase switches to the next phase [11], [12] as shown in Fig. 3 in a fixed phase order, which is due to the constraints and safety issues in the real-world setting. The other is more flexible that action is chosen from the predefined phase set [32], [57]. These two settings are tested in the experiment. More experimental details are shown in Section 5.1.3.

TABLE 1
Mathematical Notations

Notation	Description
$o_{i,t}$	Observation for agent i at time t
$a_{i,t}$	Action for agent i at time t
$r_{i,t}$	Reward for agent i at time t
$Q_{i,t}$	Q value for agent i at time t
q_l	Queue length in the lane l
n_l	Number of vehicles in the lane l
$speed_l$	Average speed of vehicles on the lane l
e_k	Observation information in the kth edge
v_i	Observation information in the ith node
l_k	Lanes number in the kth edge
$c(k)$	Edge type for the kth edge
G	$G = (V, E)$ is the constructed traffic light adjacency graph
G_t^{in}	Graph G (node's observation information is $v_{i,t}^{in}$) after node initialization at time t
G_t^{hid}	Output hidden state (node's observation information is $v_{i,t}^{hid}$) of graph LSTM at time t
G_t^{out}	Traffic light graph (node's observation information is $v_{i,t}^{out}$) after node update at time t
$v_{i,t}^d$	Updated node i's observation information vector after relation reason step d at time t
$f_e^{c_k}$	Edge encoder for edge type of c_k
Δ_t	Time interval to consider the temporal dependency
ϵ	Exploration probability
$\mathcal{D}$	Memory buffer

- *Reward* $\mathcal{R}$: $r_{i,t}$ is the immediate reward for agent i at time t. Traffic agent i is optimized to maximize the expected future return $\mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_{i,t}]$, where γ is the discount factor. The individual reward $r_{i,t}$ for agent i is $r_{i,t} = -\sum_{l=1}^{l_i} q_l$, where l_i is the number of incoming lanes connected to intersection i.
- *State Transition Probability* $\mathcal{P}$: $p(s_{t+1}|s_t, a_t)$ defines the probability of transition from state s_t to s_{t+1} when all the agents take joint action a_t.
- *Observation Probability* $\mathcal{U}$: This is the probability of observation o_{t+1}, $o_{t+1} \sim \mathcal{U}(o_{t+1}|s_{t+1}, a_t)$.

Based on the formulations above, we can formally define the problem of multi-intersection traffic light control as follows, and related mathematical notations are summarized in Table 1.

Definition 1 (Problem Definition). *Given the* traffic light adjacency graph G, *as well as the potential reward* $r_{i,t}$ *for each action* $a_{i,t}$ *made by every traffic light agent* i *at time* t, *we target at making proper decisions* $a_{i,t}$ *for each traffic light agent* i, *so that the global reward* $\sum_{i=1}^{N} r_{i,t}$ *will be maximized.*

4 Spatio-Temporal Multi-Agent Reinforcement Learning

In this section, we will introduce our Spatio-Temporal Multi-Agent Reinforcement Learning framework in detail for multi-intersection traffic light control.

4.1 Overview

The overall framework of STMARL is illustrated in Fig. 4. To be specific, we construct the graph consisting of the traffic light agents, and then use the graph block to learn *spatial structure information* on the input graph, with considering the historical traffic state as incorporating *temporal dependency*.

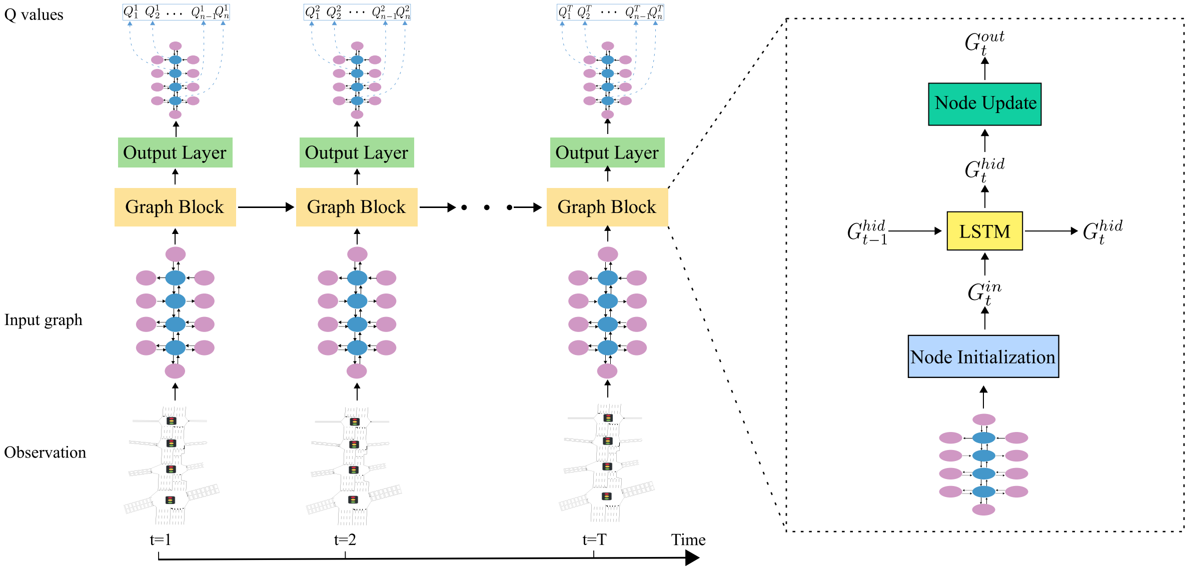


Fig. 4. Overall Q-network for spatio-temporal multi-agent reinforcement learning.

The module inside the graph block is shown in the right part of Fig. 4, which includes:

- A *node initialization* module, to get the initial node representation;
- A *recurrent neural network* variant, namely Long Short Term Memory unit (LSTM) [58] to summarize historical traffic information in hidden state to learn *temporal dependency*;
- A *node update* module, to update the state of each traffic light.

Along this line, each agent has interactions with other traffic light agents, which is beneficial for the multi-intersection traffic light control at a system level. At the same time, the problem of partial observability will be handled. In the following subsections, we will introduce all these modules in detail.

4.2 Base Multi-Agent Reinforcement Learning

First, we will briefly introduce the base multi-agent reinforcement learning method which is based on the Independent Deep Q-Networks [59], where each agent i learns the independent optimal function Q_i seperately without cooperation. Formally, for each agent i, we have the observation $o_i = \{\{q_l, n_l, w_l\}_{l=1}^{l_i}, phaseID_i\}$, where l_i is the number of incoming lanes which are connected to intersection i. We target at minimizing the following loss:

$$\mathcal{L}_i(\theta_i) = \mathbb{E}_{(o_{i,t},a_{i,t},r_{i,t},o_{i,t+1})\sim\mathcal{D}}\Big[\big(y_{i,t} - Q_i\big(o_{i,t},a_{i,t};\theta_i\big)\big)^2\Big], \tag{1}$$

$$y_{i,t} = r_{i,t} + \gamma \max_{a'} Q_i\big(o_{i,t+1},a';\theta_i^{tar}\big), \tag{2}$$

where θ_i^{tar} is the target Q-network updated periodically to stabilize training. Also, $\mathcal{D}$ is the replay buffer to store the state transitions.

For the Independent DQN, to reduce the parameters of the Q-network which scale with the number of traffic light agents, it's reasonable to share the parameters of the Q-network among agents. Specifically, the first encoder layer is separated to handle heterogeneous input information, and parameters for other layers are shared.

4.3 Learning Spatial Structure Dependency

Then, we turn to introduce the learning process inside the graph block, which considers the *spatial structure information* between the traffic light agents for better coordination. Generally, traffic light agents in the base model in Section 4.2 only learn their policy independently without explicit coordination, and observation state $o_{i,t}$ in the base model simply concatenates the traffic information in incoming lanes connected to intersection i, which ignores the spatial structure information. Therefore, a more comprehensive framework is required to leverage the spatial structure of these traffic lights for better coordination, so that the global traffic situation will be optimized.

4.3.1 Node Initialization

As the traffic information is observed in the graph edges and nodes as stated in Section 3, we first introduce the node initialization module.

Edge to Node Update. In this problem, the observed traffic information collected by the kth edge is e_k, where $e_k = \{\{q_l\}_{l=1}^{l_k}, \{n_l\}_{l=1}^{l_k}, \{speed_l\}_{l=1}^{l_k}\}$. To preserve the edge direction (e.g., four directions) to the node representation, we use the one-hot representation of the edge observation. For example, suppose there are four incoming edges connected to a traffic light agent, the one-hot representation for these four edges features can be $[e_0, \mathbf{0}, \mathbf{0}, \mathbf{0}], [\mathbf{0}, e_1, \mathbf{0}, \mathbf{0}], [\mathbf{0}, \mathbf{0}, e_2, \mathbf{0}], [\mathbf{0}, \mathbf{0}, \mathbf{0}, e_3]$. Then, to transform the raw input into embedded observation vector, edge encoder for different edge types is applied per edge to encode the collected message. The observed edge information e_k is updated as follows:

$$e'_k = f_e^{c(k)}(e_k). \tag{3}$$

To handle the heterogeneous information in the real-world and reduce parameters of edge encoder $f_e^{c(k)}$ for different edge types, separate parameters are used for the first layer of edge encoder to encode the input with different dimensions , but parameters for the other layers are shared. Specifically, we use two-layer MultiLayer Perceptron (MLP) with the Rectified Linear Units (RELU) [60] activation function.After updating the edge information, we aggregate the edges information to the receiver node as follows:

$$v^{e}_{i,t} = \sum_{k, rec_k = i} e'_k, \tag{4}$$

where rec_k is the receiver node of the kth edge.

Then nodes' initial representations are obtained by concatenating $v^{e}_{i,t}$ with the observed node feature $v_{i,t}$ at time t. Here we denote the graph with initial node values as G^{in}_t with node observation information $V = \{v^{in}_{i,t}\}^{|V|}_{i=1}$, where initial representation $v^{in}_{i,t}$ at time t is obtained as follows:

$$v^{in}_{i,t} = f_v(v^{e}_{i,t} \| v_{i,t}), \tag{5}$$

where $\|$ represents the concatenation operation and f_v is one-layer MLP. For $phaseID$ in node feature v_i, we use one-hot representation.

4.3.2 Node Update

Then, we turn to introduce the node update module to model the interaction relationship among these agents. Here, we use attention mechanism [16], [61] to leverage the spatial structure information and perform relation reasoning among these agents. Specifically, at relation reasoning step d, the input node vector consists of both the initial node vector $v^{in}_{i,t}$ and the node vector $v^{d-1}_{i,t}$ in previous relation reasoning step $d-1$

$$\hat{v}_i = [v^{in}_{i,t} \| v^{d-1}_{i,t}], \tag{6}$$

where $\|$ represents the concatenation operation.

Then, we compute the attention score α_{ij} between node i and its sender nodes $j \in \{send_k\}_{rec_k = i, k=1:|E|}$,

$$\alpha_{ij} = \frac{\exp\Big(f\big(w^T_a\left[\hat{v}_i \| \hat{v}_j\right]\big)\Big)}{\sum_{k \in \mathcal{N}_i} \exp\Big(f\big(w^T_a[\hat{v}_i \| \hat{v}_k]\big)\Big)}, \tag{7}$$

where $\|$ represents the concatenation operation. w_a is the trainable attention weight vector and f is the nonlinear activation function and here we use Exponential Linear Unit (ELU) [62] function. Then the aggregated attention information $\overline{v}_i$ from the sender nodes for node i is

$$\overline{v}_i = \sum_{j \in \{send_k\}_{rec_k = i, k=1:|E|}} \alpha_{ij} \hat{v}_j. \tag{8}$$

Finally, the node vector $v^{d}_{i,t}$ is updated based on its own information and the aggregated neighbor information

$$v^{d}_{i,t} = g([\hat{v}_i \| \overline{v}_i]), \tag{9}$$

where $\|$ represents the concatenation operation and g is one-layer MLP with RELU activation in the MLP output layer.

The above node update process is for one step relation reasoning. Multi-step relation reasoning can be performed to capture the high-order interaction among the agents. For example, if relation reasoning step is 2, the traffic light agent can aggregate information from its first-order and second-order neighbors. The output graph at time t is denoted as G^{out}_t with $V = \{v^{d}_{i,t}\}^{N}_{i=1}$.

4.4 Learning Temporal Dependency

Moreover, we attempt to learn *temporal dependency* to incorporate the historical traffic state information. As the traffic state is highly dynamic over the time, to model the temporal dependency and handle the partial observability in the POMDP problem, we use the recurrent neural network to incorporate the historical traffic information. Using recurrent neural network to summarize the history of observations is one approach to handle the partial observability in POMDP [37], [63], [64]. Specifically, we process the nodes in current input traffic state graph G^{in}_t and the last time hidden graph G^{hid}_{t-1} using Long Short Term Memory unit [58]. The output hidden graph G^{hid}_t with nodes $V = \{v^{hid}_{i,t}\}^{N}_{i=1}$. The hidden state $v^{hid}_{i,t}$ is computed as follows using LSTM as follows:

$$\begin{aligned}
i_t &= \sigma\Big(W_i\Big[v^{in}_{i,t}, v^{hid}_{i,t-1}\Big] + b_i\Big),\\
f_t &= \sigma\Big(W_f\Big[v^{in}_{i,t}, v^{hid}_{i,t-1}\Big] + b_f\Big),\\
\widetilde{C}_t &= \tanh\Big(W_C\Big[v^{in}_{i,t}, v^{hid}_{i,t-1}\Big] + b_C\Big),\\
C_t &= f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t,\\
o_t &= \sigma\Big(W_o\Big[v^{in}_{i,t}, v^{hid}_{i,t-1}\Big] + b_o\Big),\\
v^{hid}_{i,t} &= o_t \odot \tanh(C_t),
\end{aligned}$$

where, $W_f, W_i, W_C, W_o, b_f, b_i, b_C, b_o$ are parmeters of weight matrices and biases. $\odot$ represents element-wise multiplication and σ is the sigmoid function. The above update process is denoted in short as

$$v^{hid}_{i,t} = LSTM(v^{in}_{i,t}, v^{hid}_{i,t-1}). \tag{10}$$

After getting the gated hidden graph G^{hid}_t at time step t, *Node update* process in Section 4.3.2 is performed on the updated traffic input graph G^{hid}_t as is shown in the right part of Fig. 4.

4.5 Output Layer

Finally, the distributed decision for each traffic light agent is now available with the learned representations for traffic lights. Based on the above learned *spatial-temporal dependency* representation, at each time t, the decision is made for each traffic light agent. As there are two kinds of nodes: traffic light control nodes and non-control nodes shown in Fig. 2, we process for the traffic light control nodes. For these traffic light agents, we use residual connection which concatenate the initial node feature vector $v^{in}_{i,t}$ and the updated node feature vector $v^{out}_{i,t}$

$$x_{i,t} = [v^{in}_{i,t} \| v^{out}_{i,t}], \tag{11}$$

where $\|$ represents the concatenation operation.

Then, Q value for each agent i at time t is computed as follows:

$$Q_{i,t} = \phi(x_{i,t}), \tag{12}$$

where ϕ is two layer MLP with RELU activation, $Q_{i,t} \in \mathbb{R}^{|\mathcal{A}|}$ and we denote $Q_{i,t}(a)$ as the output Q value for action a.

4.6 Training

Besides, we briefly introduce the technical solution of training process. During training, we store the observations into the replay buffer $\mathcal{D}$ for experience replay [65]. As the observations are in the edges of graph G, we denote the observation at time t as $o_t = \{\{e_{k,t}\}_{k=1}^{|E|}, \{v_{i,t}\}_{i=1}^{N}\}$. We store the transition (o_t, a_t, o_{t+1}, r_t) into $\mathcal{D}$, where joint action $a_t = \{a_{i,t}\}_{i=1}^{N}$ and reward for each agent $r_t = \{r_{i,t}\}_{i=1}^{N}$. The training loss of Q-network for STMARL model is

$$\mathcal{L}(\theta) = \mathbb{E}_{\{o_t,a_t,r_t,o_{t+1}\}_{t=1}^{\Delta t} \sim \mathcal{D}} \sum_{i=1}^{N} \sum_{t=1}^{\Delta_t} (y_{i,t} - Q_{i,t}(a_{i,t};\theta))^2, \tag{13}$$

$$y_{i,t} = r_{i,t} + \gamma \max_{a'} Q_{i,t+1}\big(a';\theta^{tar}\big), \tag{14}$$

where Δt is the time interval. We use recurrent neural network over the continuous time interval Δt to learn temporal dependency as stated in Section 4.4. The influence of the temporal dependency interval is illustrated in experiment part. To stabilize training, we update the model at the end of each episode. Detailed training algorithm for Spatio-Temporal Multi-Agent Training is listed in Algorithm 1.

Algorithm 1. Spatio-Temporal Multi-Agent Training

```
Input: Traffic light adjacency graph G = (V, E).
Output: Q-network with parameter $\theta$.
Initialize the parameters of Q-network and target
  Q-network.
for epoch = 1 to max-epochs do
  Reset the environment.
  for $t = 0$ to $T$ do
    Get observation $o_t$ of the traffic light adjacency graph.
    for agent $i = 1$ to $N$ do
      Compute $v_{i,t}^{hid}$ using Eq. (10). ▷ learning temporal
        dependency
      Compute node update result $v_{i,t}^{out}$ using Eq. (9).
        ▷ learning spatial structure dependency
      Compute Q values $Q_{i,t}$ using Eq. (12).
      With probability $\epsilon$ pick random action $a_{i,t}$, else
        $a_{i,t} = max_{a'} Q_{i,t}(a')$.
    end for
    Execute joint action $a_t = \{a_{i,t}\}_{i=1}^{N}$ in the environment
      and get reward $r_t = \{r_{i,t}\}_{i=1}^{N}$ and next observation $o_{t+1}$ .
    Store transition $(o_t, a_t, o_{t+1}, r_t)$ into $\mathcal{D}$.
  end for
  for $c = 1$ to $C1$ do
    Sample a random batch of transitions over continues
      time interval $\Delta_t$: $\{o_t, a_t, r_t, o_{t+1}\}_{t=1}^{\Delta t}$ from $\mathcal{D}$.
    For each agent $i$ at time $t$ compute target:
      $y_{i,t} = r_{i,t} + \gamma \max_{a'} Q_{i,t+1}\big(a';\theta^{tar}\big)$.
    Update Q-network:
      $\theta \leftarrow \theta - \nabla_\theta \sum_{i=1}^{N} \sum_{t=1}^{\Delta_t} (y_{i,t} - Q_{i,t}(a_{i,t};\theta))^2$.
  end for
end for
```

4.7 Time Complexity Analysis

Here we analyze the time complexity of the STMARL model to show the scalability by learning spatial-temporal dependency. The following two assumptions are made that the *node initialization* can be performed concurrently for each node and for one node, the interaction with its neighbor nodes can perform separately. The neural network hidden layer size is assumed as h. Then based on STMARL model structure, the time complexity is computed as follows:

- For *node initialization*, the complexity is $d_k h + h^2 + h^2$, where d_k is the edge input feature size and the small size of $phaseID$ in node feature is ignored;
- For *node update*, the complexity to learn temporal dependency over time interval Δ_t using LSTM is $4h(h+h)\Delta_t$. For one step node update, time complexity is $4h \times 4h + 4h \times h$. Thus, time complexity for L step relation reasoning is $(8\Delta_t + 20)h^2 L$;
- For the *output layer*, time complexity is $2h^2 + h^2$.

Therefore, the overall time complexity is $O(d_k h + 5h^2 + (8\Delta_t + 20)h^2 L) \approx O(\Delta_t h^2 L)$, which scales linearly with the temporal dependency interval Δ_t and is irrelevant to the number of intersections.

5 Experiments

In this section, we conduct both quantitative and qualitative experiments to validate the effectiveness of the proposed STMARL model for multi-intersection traffic light control.

5.1 Experimental Setup

5.1.1 Synthetic Dataset

In the experiment, synthetic data is generated to test our model under various flexible traffic patterns. We generate these datasets after analyzing the real-world traffic flow data. The details are introduced as follows:

- $Unidirec_{6\times6}$: A 6×6 grid network with unidirectional traffic from West to East and South to North. The traffic flow is generated using Bernoulli distribution with probability 0.2 and the maximum number of arrival vehicles is limited to 3 in every second for stable simulation.
- $Bidirect_{6\times6}$: A 6×6 grid network with bidirectional traffic in both West-East and South-North direction. This traffic flow is generated using Bernoulli distribution with probability 0.1 and we stabilize the simulation by setting the maximum number of arrival vehicles as 4 in every second.

5.1.2 Real-World Datasets

Two real-world datasets are used here:

- $D_{Hangzhou}$:[1] A publicly available dataset for city Hangzhou, China. The road structure in this dataset is a 4×4 grid and the duration of the traffic flow is one hour.
- D_{Hefei}: This real-word dataset is collected from Hefei, China, which consists of four heterogeneous

[1] https://github.com/wingsweihua/colight/tree/master/data/Hangzhou/4_4/anon_4_4_hangzhou_real_5734.json

TABLE 2
Statistics of the Real-World Traffic Datasets

Dataset	Time Range	Arrival Rate (vehicles/300s)			
		Mean	Std	Min	Max
$D_{Hangzhou}$	-	544.83	99.19	375.00	668.00
D_{Hefei}	11/06/2018-11/12/2018	443.95	38.78	368.00	551.00

intersections as shown in Fig. 1a. The information of vehicles and roads are recorded by the camera in the nearby intersection facing the vehicles, as well as corresponding timestamps over the time period from 11/06/2018 to 11/12/2018. After analyzing these records, the trajectory for each vehicle could be captured. As shown in Table 2, the traffic arrival rate is significantly variant. For performance comparison in D_{Hefei}, we use the traffic flow during the most peak hour in one day. Further analysis of the performance of the whole day will be introduced in Section 5.2.6.

5.1.3 Simulation Setting

For synthetic datasets and $D_{Hangzhou}$, we used four phases to control the traffic movements for the intersection, i.e., WE-Straight (Going Straight from both West and East), WE-Left (Turning Left from both West and East), SN-Straight (Going Straight from both South and North), SN-Left (Turning Left from both South and North). The action of each traffic light is chosen from these four phases.

For dataset D_{Hefei}, we adopt the traffic phases currently applied in the real world during that time period as shown in Table 3 (detailed description of each phase meaning is attached in the appendix), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2020.3033782. It can be seen that there are different kinds of traffic phases for each intersection. For simplicity, we adopt the switch setting for action selection as shown in Section 3. Therefore, for each agent i, $a_{i,t} \in \{0,1\}$ indicating switch to the next phase (1) or keep current phase (0). Besides, every phase change is followed by a 3-second yellow light. One example of the executed Phase ID sequence of Traffic Light 1 (phase order is 0,1,2,3,4 as shown in Table 3) can be $0 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4$.

To simulate the traffic status under different traffic settings, we then utilized a traffic simulator called CityFlow[2] [29], [66] for large scale traffic network simulation. Both synthetic and real datasets are fed into the simulator for simulation.

5.1.4 Evaluation Metric

Following the previous researches [11], [12], [32], we adopt the commonly used *average travel time* metric to evaluate the performance of different methods. This metric is defined as the average travel time for all the vehicles traveling from their origins to destinations, which people care the most in practice. This average travel time $avgt$ is calculated as

TABLE 3
Traffic Light Phase Configurations for Different Traffic Lights in the Real-World Dataset D_{Hefei}

Traffic Light ID	Phase ID
Traffic Light 1	0, 1, 2, 3, 4
Traffic Light 2	5, 6, 7, 8
Traffic Light 3	9, 10, 11
Traffic Light 4	12, 13, 14

$$avgt = \frac{1}{N_c}\sum_{i=1}^{N_c} t_{i,ed} - t_{i,st},$$

where N_c is the toal number of cars entering this area. $t_{i,st}$ and $t_{i,ed}$ are the arrival and depart time for the ith car.

5.1.5 Implementation Details

The parameters for our model are summarized in Table 4. Particularly, each action of agent will last for 10 seconds to avoid frequent phase switch. Besides, the temporal dependency interval Δt was searched among $\{3, 5, 10, 15, 20\}$, the ϵ for ϵ-greedy policy was linearly decayed for the first 10 episodes, and the hidden size for both edge encoder, output layer and LSTM were set to 64. The activation function was searched among $\{RELU, ELU, tanh\}$ and the number of MLP layers was searched among $\{1, 2\}$. Finally, all the parameters were initialized using He initialization in [67], and then trained using Adam [68] algorithm with learning rate as 0.001 and gradient clipping value as 10.

5.1.6 Compared Methods

To validate the effectiveness of STMARL framework, several state-of-the-art methods are selected as baseline methods. There are mainly two categories: transportation methods and reinforcement learning methods.

Transportation methods are listed as follows:

- *Fixed-time Control: (Fixed-time) [8]*, which uses a predefined plan for traffic light control.
- *MaxPressure [30]*, It is the state-of-the-art transportation method, which greedily choose the phase with the maximum pressure to optimize network-level traffic light control.

The following reinforcement learning methods are compared:

- *Max-Plus Coordination for Urban Traffic Control (Max-Plus) [15]*, which uses max-plus [25] algorithm to

TABLE 4
Parameter Settings of Our Model

Parameters	Values
Memory buffer size	50 episodes
Model update step	1 episode
Target update step C	2 episodes
$C1$	3000
γ	0.99
ϵ	$1 \rightarrow 0.05$(linear decay)
Relation reasoning step d	2
batch size	16

[2] http://cityflow-project.github.io

TABLE 5
Summary of the Learning-Based Comparison Methods

Methods	Traffic Light Adjacency Graph (intersection-level)	Neighbor Attention	Temporal Dependency
Max-Plus [15]	Undirected	×	×
Neighbor RL [22]	Undirected	×	×
GCN-lane [31]	×	×	×
GCN-inter	Undirected	×	×
Colight [32]	×	✓	×
STMARL-ST	×	×	×
STMARL-T	✓	✓	×
STMARL-S	✓	×	✓
STMARL	✓	✓	✓

learn the optimal joint action based on a constructed undirected coordination graph.

- *Neighbor RL [22]*, which concatenate their neighbor's observation information into their own state representation. It does not distinguish different neighboring states.
- *GCN-lane [31]*, which uses graph convolution neural network to extract traffic features of distant roads. Its graph is constructed on lane-level where each lane is regarded as a node and vehicle traffic movement connected two lanes is represented as an edge.
- *GCN-inter*, which constructs its graph on intersection-level where each intersection is regarded as a node and the road connected two intersections is represented as an edge.
- *Colight [32]*, which is a recent method using graph attention network for multi-intersection traffic light control. This method determines the agent's neighbors using rules and the number of each agent's neighbors is predefined.[3]

Besides, the following variations of the proposed STMARL model are compared:

- *STMARL-ST*, which is the base independent DQN method with shared parameters among the agents. Specifically, the first encoder layer is separated to handle heterogeneous input information, and parameters for other layers are shared.
- *STMARL-T*, which does not learn temporal dependency and only incorporates the spatial structure information for iterative relational reasoning.
- *STMARL-S*, which only learns the temporal dependency to incorporate the historical traffic information without incorporating spatial structure dependency.

For better illustration, we summarize the characteristics of learning-based methods in Table 5. For the reinforcement learning methods, we trained the model for 100 episodes and then tested with epsilon $\epsilon = 0$.

5.2 Experimental Results

5.2.1 Overall Result

In this section, we compared the proposed method with the baseline methods on both synthetic and real-world datasets. The performance is shown in Table 6. We can observe that our proposed STMARL method significantly outperforms all the baseline methods in all datasets.

For the comparison of performance across different datasets, we observed that the performance gap becomes significantly larger when the traffic pattern changes from synthetic to real. For example, STMARL outperforms the best baseline by 20.6 percent in dataset D_{Hefei}.

Compared with transportation methods, we found that the gap between STMARL and transportation methods becomes larger when traffic pattern changes from synthetic to real. This phenomenon demonstrates the effectiveness of the reinforcement learning methods which change traffic phases adaptively to optimize the long term traffic situation.

Besides, our STMARL model significantly outperforms all the reinforcement learning baseline methods. We observed that Neighbor RL performs well under synthetic unidirectional or bidirectional traffic but fails in the large scale real-world traffic flow especially for $D_{Hangzhou}$. The reason may be that Neighbor RL only consider the one-hop neighbor relationship as well as does not consider the weight of their neighbors to respond to the real dynamic traffic flow. It can be also seen that STMARL outperforms Colight by a large margin over all the datasets. This observation indicates that it is effective to model the cooperation structure by leveraging the constructed directional traffic light adjacency graph as well as the temporal dependency. On the contrary, Colight determines each agent's neighbors by rules, and the number of neighbors is fixed, which also ignores modeling traffic flow direction on the graph. What's more, compared with these two GCN based methods, STMARL is superior in performance. As STMARL constructed the graph on the intersection level, it is more effective to model the relationship among traffic lights than the lane-level graph used by GCN-lane. The reason that GCN-lane fails to learn well in large scale road networks may also be due to its significantly increased graph complexity with a large road network such as the synthetic 6×6 road network. When building its graph on the intersection level, GCN-inter performs better than GCN-lane but fails to learn the dynamic real-world traffic flow as it treats the neighbors equally. These comparisons clearly demonstrate the effectiveness of STMARL by collectively learning spatial-temporal dependency based on the directional traffic light adjacency graph.

5.2.2 Ablation Study

The ablation study of the model component is shown in Table 7. We can find that STMARL constantly outperforms all the model variants over the datasets. Particularly, incorporating spatial structure dependency boosts performance more than the temporal dependency in most of the datasets. This demonstrates that learning spatial structure dependency is more important for cooperation among traffic lights to improve performance. Fig. 5 illustrates the training curves of these model variants across different datasets. It can be observed that incorporating spatial structure dependency greatly improved the convergence speed. Adding temporal dependency will further accelerate convergence

[3] Colight is tested based on the code https://github.com/wingsweihua/colight

TABLE 6
Performance Comparison on the Synthetic Data and Real-World Data w.r.t. Average Travel Time (in Seconds, the Lower the Better)

Methods	$Unidirec_{6\times6}$	$Bidirect_{6\times6}$	$D_{Hangzhou}$	D_{Hefei}
Fixed-time	608.34	481.35	572.15	153.63
MaxPressure	289.04	227.95	475.89	92.12
Max-Plus	891.68	1127.19	1123.55	111.56
Neighbor RL	213.98	182.20	331.95	80.45
GCN-lane	611.28	227.31	690.36	130.64
GCN-inter	276.85	213.36	401.25	98.41
Colight	273.92	194.84	394.10	249.70
STMARL	$\mathbf{205.34}^*$	$\mathbf{180.31}^*$	$\mathbf{319.14}^*$	$\mathbf{63.86}^*$

'' indicates the improvement of STMARL over the best baseline is significant based on paired t-test at the significance level of $p < 0.01$. Result on D_{Hefei} is averaged over seven days.*

and improve performance. These quantitative results clearly demonstrate the effectiveness of jointly learning spatial structure information and temporal dependency for multi-intersection traffic light control.

5.2.3 Influence of Temporal Dependency Interval Δt

In this section, we show the sensitiveness of temporal dependency interval Δt and the scalability of STMARL with different Δt.

Sensitiveness. Fig. 6a shows the performance of STMARL model with different temporal dependency interval Δt. We observe that STMARL achieves the best performance when $\Delta t = 10, 5, 3, 20$ for $Unidirec_{6\times6}$, $Bidirect_{6\times6}$, $D_{Hangzhou}$ and D_{Hefei} respectively. These results indicate that a relatively medium time interval Δt should be more appropriate to learn the temporal dependency.

Scalability. Fig. 6b shows the running time of STMARL model under different temporal dependency interval across different datasets for 100 episodes. It can be observed that the training time of the STMARL scale almost linearly with the increasing Δt under different scale road networks. On the real-world datasets $D_{Hangzhou}$ and D_{Hefei}, the training of STMARL is efficient across different Δt, which demonstrates the scalability of STMARL on large scale real-world traffic light control.

5.2.4 Influence of Hidden Layer Size h

Fig. 9 shows the performance of STMARL with different hidden layer seize h across all datasets. We can observe that STMARL achieves the best performance when the hidden layer size is 64 for all the datasets, which indicates a medium time complexity according to Section 4.7.

TABLE 7
Ablation Study of the Model Component

Methods	$Unidirec_{6\times6}$	$Bidirect_{6\times6}$	$D_{Hangzhou}$	D_{Hefei}
STMARL-ST	229.75	194.42	329.97	71.94
STMARL-T	214.68	180.91	322.17	66.02
STMARL-S	218.09	181.13	329.64	65.58
STMARL	$\mathbf{205.34}^*$	$\mathbf{180.31}^*$	$\mathbf{319.14}^*$	$\mathbf{63.86}^*$

'' indicates the improvement of STMARL over each ablation model is significant based on paired t test with $p < 0.01$.*

5.2.5 Fairness of STMARL Method

In this section, we discuss the fairness of STMARL method among vehicles. As shown in Table 8, the standard deviation of the travel time among vehicles for STMARL method is the smallest compared with the baseline methods in all the datasets. This result demonstrates that our STMARL method is fair among vehicles.

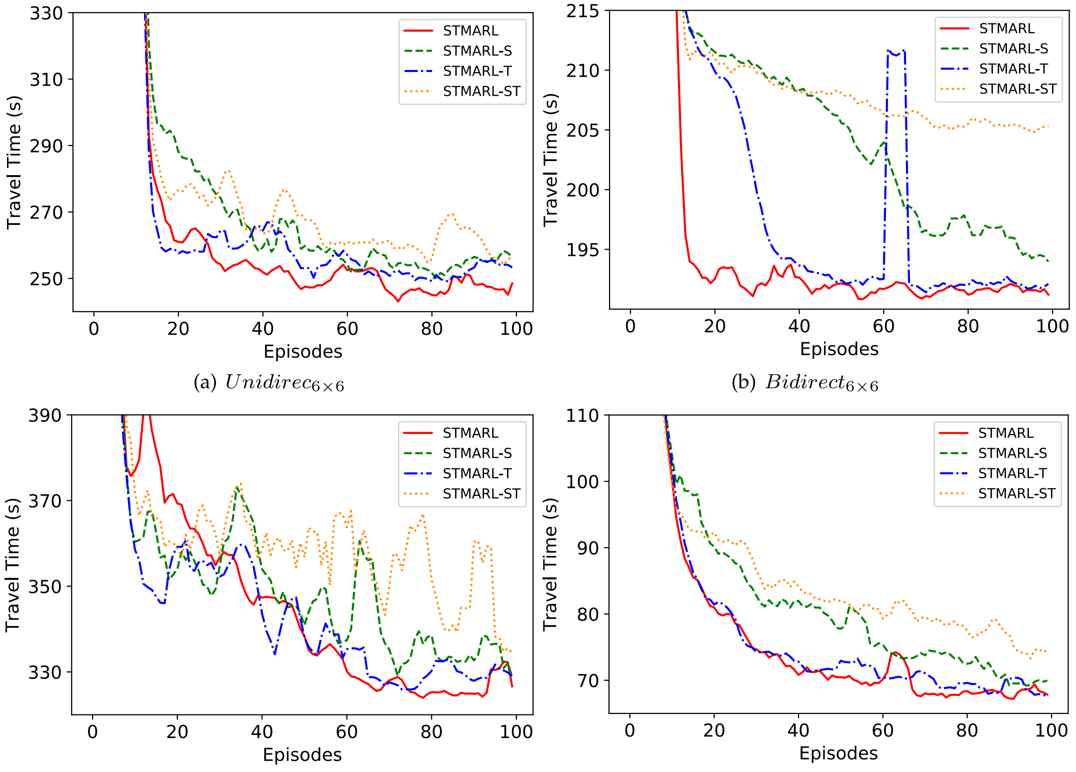
5.2.6 Qualitative Study

In this section, we provide further analysis of the learned attention weights to the adjacent nodes and the emergence of the green wave learned by the STMARL model in the real-world dataset D_{Hefei} for full-day analysis.

Interpretation of Attention Weights. In this section, we analyze the learned attention weights for the traffic light agent and take the Traffic Light agent 2 as an example. Figs. 7c and 7d shows the learned attention weights in four incoming edges from its neighbors on Tuesday and Saturday. We also show the average number of approaching vehicles in corresponding edges in Figs. 7a and 7b. It can be observed that the learned attention weights keep pace with the dynamic number of vehicles in the corresponding edges. For example, in different peak hours, the attention weights in the corresponding edge directions also become larger.

Moreover, in Fig. 7c, the attention weight of the South-North direction is the largest compared to the other three directions in most cases, which corresponds to the largest arriving vehicle numbers as shown in Fig. 7a. Similar rule could be found between Figs. 7d and 7b. Therefore, the larger attention weights make the Traffic Light agent 2 care more about the downstream traffic situation which may spill into intersection 2. As a result, the Traffic Light agent 2 was influenced by the decision of Traffic Light agent 1. Along this line, the coordination among Traffic Light agent 2 and agent 1 is crucial so as not to cause severe traffic congestion to Intersection 2 due to the large traffic flow from intersection 1. Therefore, the larger attention weights in the edge direction may indicate that it's more necessary to coordinate among these two agents.

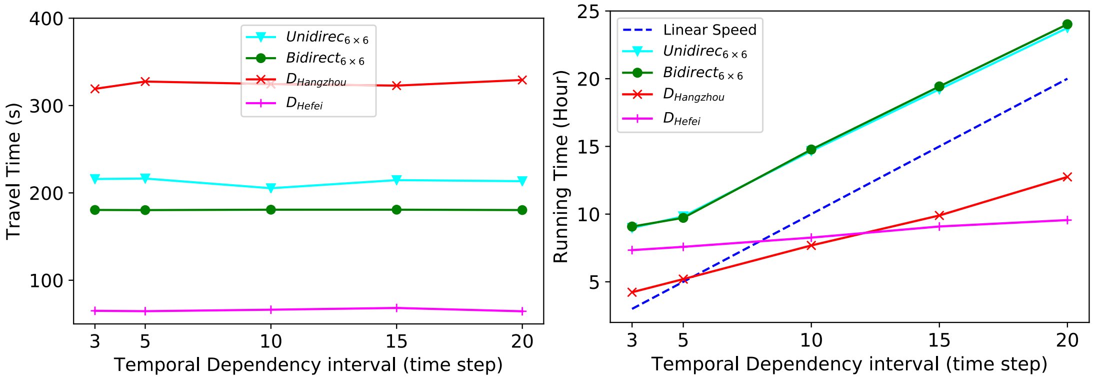
Cooperation for Green Wave. The green wave occurs when a series of traffic lights are coordinated to allow continuous traffic flow over several intersections along one main direction.



(a) $Unidirec_{6\times6}$ (b) $Bidirect_{6\times6}$

(c) $D_{Hangzhou}$ (d) D_{Hefei}

Fig. 5. Training curve of our model variants across different datasets. Curves are smoothed with a moving average of five points.



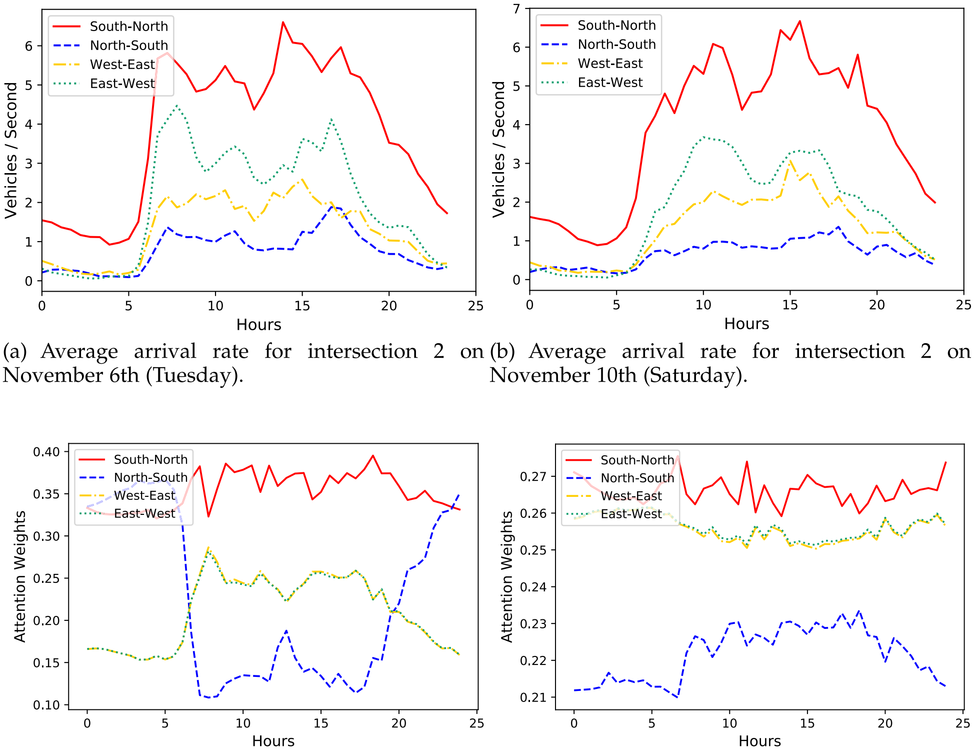
(a) Performance of STMARL model with different Δt. (b) Running time of STMARL model with different Δt.

Fig. 6. Influence of temporal dependency interval Δt.

Therefore, it can be used to test the coordination mechanism learned by multiple traffic lights. Fig. 8 shows the dynamics of traffic light phase learned by our model (Figs. 8a, 8b, and 8c) and the corresponding number of approaching cars along South to North direction (Figs. 8d, 8e, and 8f) which shows the emergence of *green wave* phenomenon. It can be observed that from Figs. 8a, 8b, and 8c, in these three time periods, there exists a green wave, i.e., green arrow, where all the four traffic light agents coordinated their traffic phases (current green phase in South-North direction) to allow fast traveling for the approaching cars. Figs. 8d, 8e, and 8f shows that the green wave significantly accelerates the traffic flow by reducing the maximum number of vehicles approaching one intersection (e.g., the intersection controlled by Traffic Light 1). It also shows the peak of vehicle number moves from intersection 1 to intersection 2 along the green wave direction, which indicates the fast-moving of traffic flow. Therefore, the green wave demonstrates that the STMARL model can learn coordination policy to reduce traffic congestion at an integral level.

TABLE 8
Standard Deviation of the Travel Time Among Vehicles in an Area

Methods	$Unidirec_{6\times6}$	$Bidirect_{6\times6}$	$D_{Hangzhou}$	D_{Hefei}
Fixed-time	167.28	151.09	520.96	276.32
MaxPressure	73.97	50.37	447.54	224.11
Max-Plus	804.43	975.38	1179.10	211.17
Neighbor RL	39.65	26.97	226.66	191.76
GCN-lane	514.76	36.05	740.16	231.20
GCN-inter	62.74	35.88	316.33	214.61
Colight	273.92	194.84	394.10	655.38
STMARL	**36.10**	**26.14**	**208.64**	**174.47**



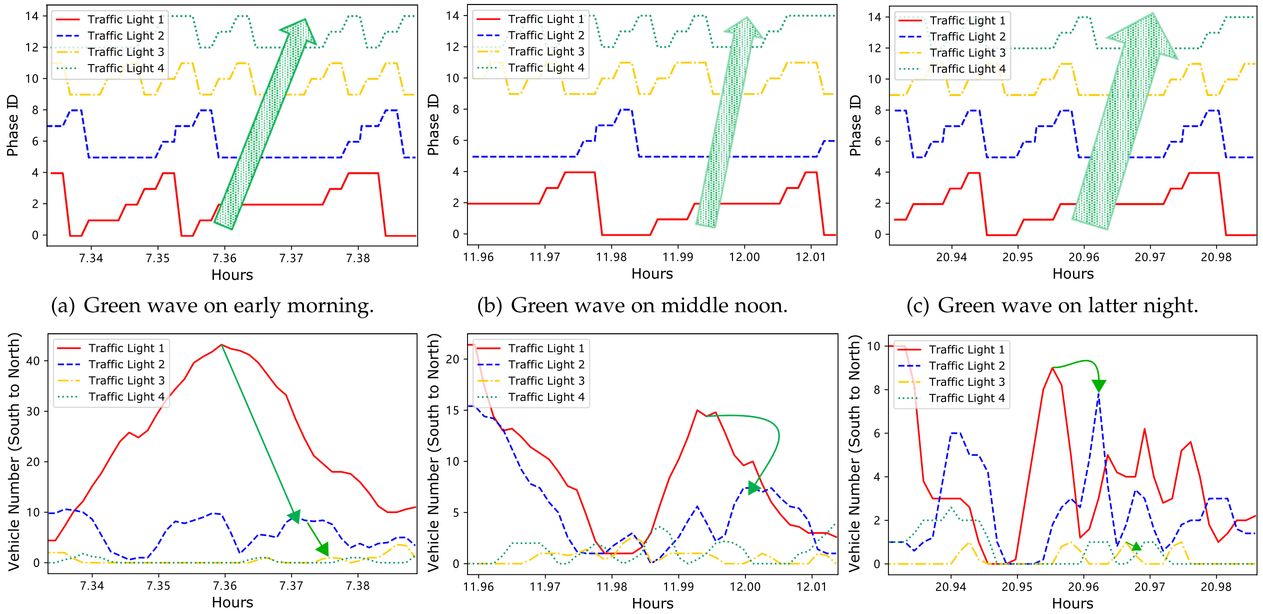
(a) Average arrival rate for intersection 2 on November 6th (Tuesday).

(b) Average arrival rate for intersection 2 on November 10th (Saturday).

(c) Learned attention weights for intersection 2 on November 6th (Tuesday).

(d) Learned attention weights for intersection 2 on November 10th (Saturday).

Fig. 7. Average arriving vehicles ((a), (b)) and learned attention weights ((c), (d)) by STMARL model on the four incoming edges (e.g., South-North means the edge direction is from South to North connected to this intersection) of intersection 2 (node TL 2 in Fig. 2) On November 6th (Tuesday) and November 10th (Saturday), 2018, Hefei.



(a) Green wave on early morning.

(b) Green wave on middle noon.

(c) Green wave on latter night.

(d) Number of vehicles approaching on early morning.

(e) Number of Vehicles approaching on middle noon.

(f) Number of vehicles approaching on latter night.

Fig. 8. The emerging green waves show in different time periods on November 6th, 2018, Hefei, in which the waves pointed by the green arrows indicate the green phases of traffic lights coordinated in South to North direction.

6 Conclusion

In this paper, we proposed the Spatio-Temporal Multi-Agent Reinforcement Learning model for multi-intersection traffic light control. The proposed STMARL method can leverage the spatial structure in the real world to facilitate coordination among multiple traffic lights. Moreover, it also considers the historical traffic information for current decision making. Specifically, we first construct the traffic light adjacency graph based on the spatial structure among traffic

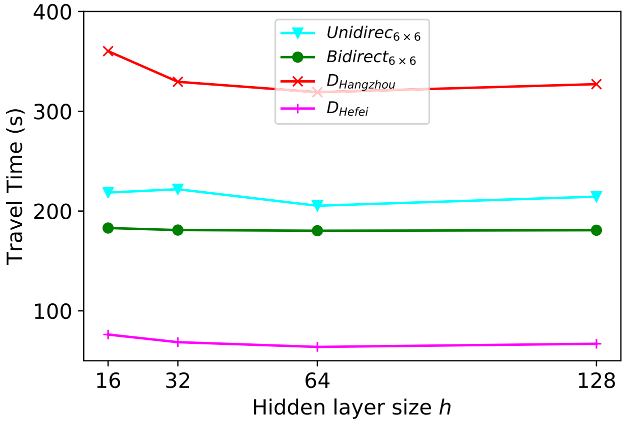


Fig. 9. Influence of hidden layer size h.

lights. Then, historical traffic records will be integrated with current traffic status via Recurrent Neural Network structure. Moreover, based on the temporally-dependent traffic information, we design a Graph Neural Network based model to represent relationships among multiple traffic lights, and the decision for each traffic light will be made in a distributed way by deep Q-learning method. Experiments on both synthetic and real-world datasets have demonstrated the effectiveness of our STMARL framework, which also provides an insightful understanding of the influence mechanism among multi-intersection traffic lights.

Acknowledgments

This work was supported in part by Grants from the National Key Research and Development Program of China under Grant No.2018YFB1402600, and the National Natural Science Foundation of China under Grant Nos.91746301, 71531001, 61703386, U1605251.

References

[1] CityLab, "Traffic's mind-boggling economic toll," 2018. [Online]. Available: https://www.citylab.com/transportation/2018/02/traffics-mind-boggling-e conomic-toll/552488/

[2] T. Xu, H. Zhu, H. Xiong, H. Zhong, and E. Chen, "Exploring the social learning of taxi drivers in latent vehicle-to-vehicle networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1804–1817, Aug. 2020.

[3] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[4] C. Wu, A. Pozdnukhov, and A. M. Bayen, "Block simplex signal recovery: Methods, trade-offs, and an application to routing," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1547–1559, Apr. 2020.

[5] T. Cabannes, M. Sangiovanni, A. Keimer, and A. M. Bayen, "Regrets in routing networks: Measuring the impact of routing apps in traffic," *ACM Trans. Spatial Algorithms Syst.*, vol. 5, pp. 1–19, 2019.

[6] S. Li, J. Zhou, T. Xu, H. Liu, X. Lu, and H. Xiong, "Competitive analysis for points of interest," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1265–1274.

[7] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 1186–1193.

[8] A. J. Miller, "Settings for fixed-cycle traffic signals," *J. Oper. Res. Soc.*, vol. 14, pp. 373–386, 1963.

[9] B. Yin, M. Dridi, and A. El Moudni, "Traffic network micro-simulation model and control algorithm based on approximate dynamic programming," *IET Intell. Transport Syst.*, vol. 10, pp. 186–196, 2016.

[10] S.-B. Cools, C. Gershenson, and B. DHooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in Applied Self-Organizing Systems*. Berlin, Germany: Springer, 2013, pp. 45–55.

[11] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2496–2505.

[12] 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*. Berlin, Germany: Springer, 2016, pp. 47–66.

[13] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2746–2754.

[14] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[15] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2008, pp. 656–671.

[16] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[17] 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.

[18] M. A. Khamis and W. Gomaa, "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 134–151, 2014.

[19] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *J. Transp. Eng.*, vol. 129, pp. 278–285, 2003.

[20] S. El-Tantawy and B. Abdulhai, "An agent-based learning towards decentralized and coordinated traffic signal control," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 665–670.

[21] M. Steingrover *et al.*, "Reinforcement learning of traffic light controllers adapting to traffic congestion," in *Proc. 17th Belgium-Netherlands Conf. Artif. Intell.*, 2005, pp. 216–223.

[22] I. Arel, C. Liu, T. Urbanik, and A. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transport Syst.*, vol. 4, pp. 128–135, 2010.

[23] S. G. Rizzo, G. Vantini, and S. Chawla, "Time critic policy gradient methods for traffic signal control in complex and congested scenarios," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1654–1664.

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

[25] J. R. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *J. Mach. Learn. Res.*, vol. 7, pp. 1789–1828, 2006.

[26] B. Bakker, S. Whiteson, L. Kester, and F. C. Groen, "Traffic light control by multiagent reinforcement learning systems," in *Interactive Collaborative Information Systems*. Berlin, Germany: Springer, 2010, pp. 475–510.

[27] M. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 1151–1158.

[28] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[29] H. Wei *et al.*, "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1290–1298.

[30] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," in *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Berlin, Germany: Springer, 2013, pp. 27–66.

[31] T. Nishi, K. Otaki, K. Hayakawa, and T. Yoshimura, "Traffic signal control based on reinforcement learning with graph convolutional neural nets," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 877–883.

[32] H. Wei *et al.*, "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1913–1922.

[33] A. Bazzi, A. Zanella, and B. M. Masini, "A distributed virtual traffic light algorithm exploiting short range V2V communications," *Ad Hoc Netw.*, vol. 49, pp. 42–57, 2016.

[34] M. Ferreira and P. M. d'Orey, "On the impact of virtual traffic lights on carbon emissions mitigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 284–295, Mar. 2012.

[35] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 66–83.

[36] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi-Agent Syst.*, vol. 11, pp. 387–434, 2005.

[37] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2137–2145.

[38] S. Sukhbaatar *et al.*, "Learning multiagent communication with backpropagation," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2244–2252.

[39] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for localization-of-things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov. 2019.

[40] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, Jul. 2018.

[41] P. Sharma, A.-A. Saucan, D. J. Bucci, and P. K. Varshney, "Decentralized Gaussian filters for cooperative self-localization and multi-target tracking," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5896–5911, Nov. 2019.

[42] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Experiments with cooperative control of underwater robots," *The Int. J. Robot. Res.*, vol. 28, pp. 815–833, 2009.

[43] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 509–522, Jun. 2011.

[44] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *NIPS Deep Learn. Workshop*, 2013.

[45] A. Tampuu *et al.*, "Multiagent cooperation and competition with deep reinforcement learning," *PloS One*, vol. 12, 2017, Art. no. e0172395.

[46] E. Zawadzki, A. Lipson, and K. Leyton-Brown, "Empirically evaluating multiagent learning algorithms," 2014, *arXiv:1401.8074*.

[47] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[48] J. Jiang, C. Dun, T. Huang, and Z. Lu, "Graph convolutional reinforcement learning," in *Proc. 8th Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HkxdQkSYDB

[49] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[50] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv: 1806.01261*.

[51] P. Battaglia *et al.*, "Interaction networks for learning about objects, relations and physics," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4502–4510.

[52] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *Proc. 4th Int. Conf. Learn. Representations*, 2016. [Online]. Available: http://arxiv.org/abs/1511.05493

[53] Z. Zhi-Hua, "Abductive learning: Towards bridging machine learning and logical reasoning," *Sci. China Inf. Sci.*, no. 7, 2019, Art. no. 21.

[54] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[55] T. Wang, R. Liao, J. Ba, and S. Fidler, "NerveNet: Learning structured policy with graph neural networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=S1sqHMZCb

[56] V. Zambaldi *et al.*, "Deep reinforcement learning with relational inductive biases," in *Proc. 7th Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HkxaFoC9KQ

[57] M. Wiering, J. Vreeken, J. Van Veenen, and A. Koopman, "Simulation and optimization of traffic in a city," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 453–458.

[58] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[59] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.

[60] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.

[61] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[62] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by Exponential Linear Units (ELUs)," in *Proc. 4th Int. Conf. Learn. Representations*, 2016. [Online]. Available: http://arxiv.org/abs/1511.07289

[63] M. Hausknecht and P. Stone, "Deep recurrent Q-Learning for partially observable MDPs," *AAAI Fall Symp.*, Arlington, Virginia, USA, Nov. 12–14, pp. 29–37, 2015.

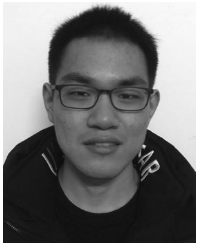
[64] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *NIPS Deep Reinforcement Learn. Workshop*, 2015.

[65] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, 2015, Art. no. 529.

[66] H. Zhang *et al.*, "CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *Proc. World Wide Web Conf.*, 2019, pp. 3620–3624.

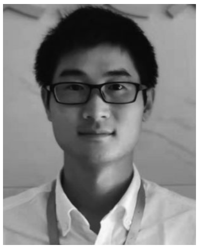
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

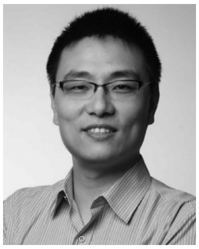


Yanan Wang received the BE degree from the South China University of Technology (SCUT), Guangzhou, China, in 2017, the ME degree from the University of Science and Technology of China (USTC), Hefei, China, in 2020. He is currently an incomming PhD degree at Eller College of Management, the University of Arizona, Tucson, Arizona. His main research interests include reinforcement learning, data mining and natural language processing.

Tong Xu (Member, IEEE) received the PhD degree from the University of Science and Technology of China (USTC), Hefei, China, in 2016. He is currently working as an associate professor at the Anhui Province Key Laboratory of Big Data Analysis and Application, USTC. He has authored more than 60 journal and conference papers in the fields of social network and social media analysis, including the *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Multimedia*, KDD, AAAI, ICDM, etc.



Xin Niu is currently working in iFLYTEK, mainly responsible for data mining and urban big data analysis.



Chang Tan is currently a general manager of iFLYTEK Intelligent Transportation Bussiness Department. His general area of research interests include data mining, with a focus on solving urban traffic problems using big data.

Enhong Chen (Senior Member, IEEE) received the PhD degree from the University of Science and Technology of China, China. He is currently a professor and vice dean at the School of Computer Science at the University of Science and Technology of China (USTC), China. His research interests include data mining and machine learning, social network analysis and recommender systems. He has published more than 100 papers in refereed conferences and journals, including the *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Mobile Computing*, KDD, ICDM, NIPS, and CIKM. He was on program committees of numerous conferences including KDD, ICDM, SDM. He received the Best Application Paper Award on KDD-2008, the Best Student Paper Award on KDD-2018 (Research), the Best Research Paper Award on ICDM-2011 and Best of SDM-2015. His research is supported by the National Science Foundation for Distinguished Young Scholars of China.

Hui Xiong (Fellow, IEEE) received the PhD degree from the University of Minnesota (UMN), Minneapolis, Minnesota. He is currently a full professor at the Rutgers, the State University of New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He is a co-editor-in-chief of Encyclopedia of GIS, an associate editor of the *IEEE Transactions on Big Data (TBD)*, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, and *ACM Transactions on Management Information Systems (TMIS)*. He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a program co-chair for the IEEE 2013 International Conference on Data Mining (ICDM), a general co-chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a program co-chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an ACM distinguished scientist.