Sensors and Materials, Vol. 34, No. 2 (2022) 545–562
MYU Tokyo

S & M 2827

NCCLight: Neighborhood Cognitive Consistency for Traffic Signal Control

Yan Kong* and Shan Cong

Computer and Software School, Nanjing University of Information Science and Technology,
219# Ningliu Street, Nanjing, Jiangsu 210044, PR China

(Received July 7, 2021; accepted November 25, 2021; online published January 24, 2022)

Keywords: multi-agent reinforcement learning, traffic signal control, neighborhood cognitive consistency

Multi-agent reinforcement learning (MARL) is gradually becoming an attractive research field of adaptive traffic signal control (ATSC). Nevertheless, in a multi-agent environment, some inherent disadvantages exist, such as the partial observability and non-stationarity caused by the constantly changing decision-making strategies of agents, which have been extensively researched but remain challenging. Herein, NCCLight, which is a fully scalable decentralized MARL model built around an independent advantage actor-critic (IA2C) under the background of ATSC, is rationally designed and validated to offer a feasible approach to realizing communication and coordination between multiple agents. In addition, guided by cognitive consistency theory, the constraint of neighborhood cognitive consistency (NCC) is constructed to achieve communication and coordination between multiple agents. More significantly, cognitive consistency theory is employed in MARL for ATSC for the first time, which is validated by a large number of experiments on both real and synthetic data. We hope that this work can serve as a pioneering reference owing to the better performance of NCCLight than of the most advanced ATSC based on MARL.

1. Introduction

An adaptive traffic signal control (ATSC) system can control traffic signals according to the real-time traffic state, markedly improving the traffic capacity and traffic quality of intersections. To reduce road congestion, multiple models have been developed, and ATSC systems are considered the most effective means to alleviate traffic congestion. Reinforcement learning (RL) technology has received considerable attention in the artificial intelligence field as well as ATSC.(1,2) Unfortunately, current traffic signal control systems follow predetermined signal schemes owing to insufficient training data obtained from sensors. Furthermore, it is urgently necessary but still challenging to change signal schemes and learn from results; such trial and error is the core idea of RL.(3) In an RL system, agents implement strategies according to the current environment, then adjust the strategies according to the feedback of the environment, and such a principle is used to form the ATSC system.(4)

*Corresponding author: e-mail: kongyan4282@163.com
https://doi.org/10.18494/SAM3507

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

Recently, centralized RL techniques have been proposed to coordinate traffic signals. For example, Prashanth and Bhatnagar directly trained a central agent to order the actions for all intersections, and Kuyer *et al.* utilized centralized coordination over global joint actions.(5,6) However, it is not feasible to take advantage of these centralized models to control multiple traffic signal agents in a large-scale traffic network. First, these centralized models need to collect the state information of all agents in the traffic network and then provide them to the agents as the global state. Secondly, the dimension of the agents' joint action space increases exponentially with the number of intersections. Therefore, these centralized models have huge state and action spaces, which will inevitably induce high delay and failure rates in practice.

To effectively speed up the solution to the problem, independent RL models are employed, where each intersection is regulated by an RL agent with its own strategy. Even though this model can overcome the scalability issue, many other issues arise. The major disadvantage is the non-stationarity of multi-agent environments. Even if the same agent performs the same action under the same state, both the reward signal obtained and the new observed state are still uncertain, while the environment is affected by the actions of other agents. Therefore, the Markov hypothesis of the single-agent RL algorithm is not valid in a multi-agent algorithm. Nguyen *et al.* systematically summarized ways to deal with the non-stationarity from the viewpoint of communication and coordination.(7) Through their designed communication and coordination mechanism, agents can exchange information via their observations and adopt strategies to stabilize their training. In addition, the observation of a single agent is usually only part of the entire environment, namely, a multi-agent environment is partially observable, which will significantly affect the performance of training. To solve the above problems, Chu *et al.* and Dresner and Stone constructed a promising platform for cooperation among agents.(8,9) Buşoniu *et al.*,(10) Mao *et al.*,(11) and Arel *et al.*(12) added neighboring states, neighbors' hidden states, and downstream information into states, respectively. These models seem to work but do not completely solve the problem. In these models, the intersection agent only connects the traffic conditions of adjacent intersections with its own traffic conditions according to static prior geographical knowledge, and the congestion of intersections is updated dynamically.(13–15)

To thoroughly solve the above problems, cognitive consistency theory is investigated in this paper. This theory is a social psychological theory that mainly explores how people adjust their original attitudes after receiving new information. In multi-agent systems, cognitive consistency plays a significant role because people are used to seeking balance and harmony.(16-19) Neighborhood cognitive consistency (NCC) was first proposed by Mao *et al.*, who designed personalized coordination strategies based on the consistency of neighbor-specific cognition and the difference in individual-specific cognition, as well as conducted extensive experiments on multiple challenging tasks. Moreover, only cognitive consistency between neighbors is essential, which results from humans interacting directly with their neighbors.(20)

Taking the above-mentioned issues together, in this paper, we rationally design a mechanism for multi-agent cooperation and communication based on NCC under the background of ATSC. Firstly, individual cognition is defined as an agent's cognition of local traffic congestion, including traffic congestion at adjacent intersections. NCC is defined as an agent's cognition of traffic congestion in the neighborhood. On this basis, we assume that each agent in the same neighborhood possesses a common hidden cognitive variable, and agents make their neighborhood cognitive representation consistent with the common hidden cognitive variable by introducing a normalizing flow under a model of stochastic gradient variational inference. In this way, all agents in the same neighborhood will eventually realize NCC, which can effectively solve the problem of partial observability and non-stationarity.

To design a stable and robust algorithm for ATSC, we extend the idea of NCC to an independent advantage actor-critic (IA2C). Specifically, a multi-intersection environment is first modeled as a traffic graph, and then a graph convolutional network is applied to draw a high-level representation from the joint observation of all adjacent agents to give each local agent more information about regional traffic congestion. Secondly, we extract the neighborhood cognitive representation from such a high-level representation and exploit the NCC constraints to achieve more efficient cooperation among agents. To corroborate these hypotheses, NCCLight, our proposed multi-agent reinforcement learning (MARL) model, is evaluated using two synthetic large traffic grids and three real traffic networks. Numerical experiments show that NCCLight is superior to other advanced algorithms in terms of robustness and optimality. The technological innovations of this work are as follows.

- We are in the vanguard of researching the communication and cooperation of multiple intersections based on NCC, through which neighborhood agents can realize system-level cooperation.
- An innovative MARL model based on IA2C is rationally devised, which adopts graph convolutional networks to extract high-level information representations, and implements cooperation and communication among multiple agents by adding NCC constraints.

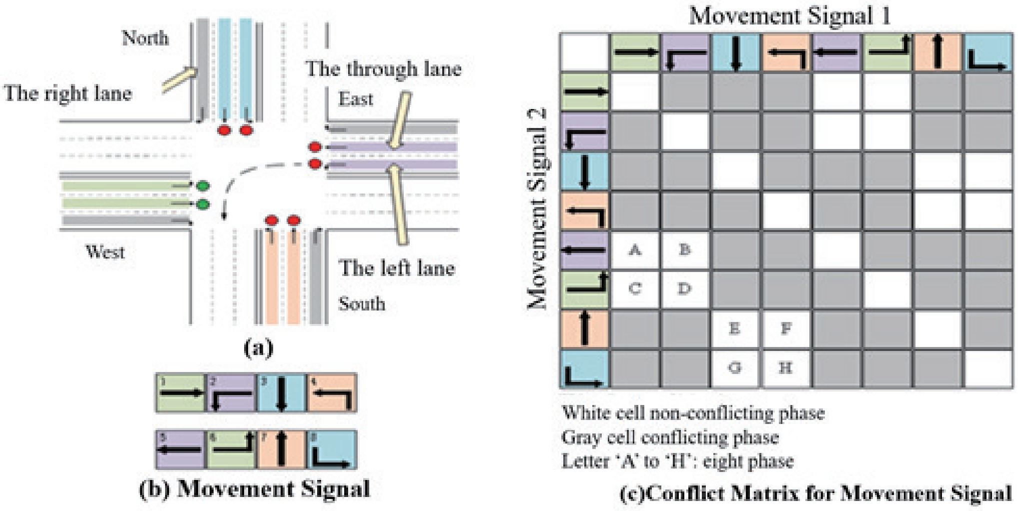
2. Problem Statement

We investigate traffic signal control in multi-intersection scenarios. In most cases, the intersection scene is determined by the movement signal and the phase setting, which are explained as follows.

Movement signal: a movement signal is defined on the basis of the traffic movement. For the intersection shown in Fig. 1(a), right-turning vehicles can pass without being affected by signals. A total of eight motion signals are used, as shown in Fig. 1(b).

Phase setting: a phase is a combination of movement signals, and each phase controls two non-conflicting traffic movements; thus, there are theoretically eight phases in an intersection, as shown in Fig. 1(c). A gray cell indicates that the two corresponding movements conflict with each other, that is, they cannot both be set to green simultaneously. For example, signals 1 and 2 cannot both be set to green at the same time. In contrast, a white cell indicates that the two corresponding movements are not in conflict. Here, we only consider the paired-signal phases [labels A to H in Fig. 1(c)] to increase the productivity compared with the use of single-signal phases.

In a traffic network, an intersection is defined as a node and the road segment between two adjacent intersections is defined as an edge. Thus, an entire traffic network can be defined as a complex network composed of a number of nodes and edges. According to graph theory, the


North
The right lane
The through lane
East
West
The left lane
South
(a)
1
2
3
4
5
6
7
8
(b) Movement Signal
Movement Signal 1
Movement Signal 2
White cell non-conflicting phase
Gray cell conflicting phase
Letter 'A' to 'H': eight phase
(c)Conflict Matrix for Movement Signal

	→	↲	↓	↰	←	↱	↑	↳
→	□	■	■	■	□	□	■	■
↲	■	□	■	■	□	□	■	■
↓	■	■	□	■	■	■	□	□
↰	■	■	■	□	■	■	□	□
←	A	B	■	■	□	■	■	■
↱	C	D	■	■	■	□	■	■
↑	■	■	E	F	■	■	□	■
↳	■	■	G	H	■	■	■	□

Fig. 1. (Color online) Traffic movement and traffic signal phases.

traffic network is abstracted as a graph $\vec{G} = (V, E)$. Specifically, $V = \{V_i\}$ represents the set of nodes in the graph and $E = \{seg_{ij}\}$ represents the set of edges, where seg_{ij} refers to the edge from node n_i to node n_j. The neighboring nodes of node i are represented as $N(i)$, and each node $j \in N(i)$ is within the neighborhood of node i.

After modeling the traffic network as a graph, we study the traffic light control problem at multiple intersections from the perspective of MARL. Roughly, we regard each intersection as an agent and coordinate this group of agents to maximize the overall reward. From this perspective, the multi-intersection traffic light control problem can be defined as a Markov decision process (MDP) with a finite number of steps. In addition, considering that the recorded traffic information may be incomplete and inaccurate, the MDP problem is further extended to a partially observable Markov decision process (POMDP), which can be defined as a four-tuple $\{N, S, O, A, P, R\}$, where N is the number of agents and the other variables are defined as follows.

Hidden full state space *S* and partial observation space *O*: usually, agents can only have local views of the system state S, which results in only a partial observation space $O_t = \{o_{i,t}\}_{i=1}^{N}$ being accessible at time t. In this work, we define the incomplete state information observed by agent i at time t as $o_{i,t}$, which is composed of the current traffic light phase of intersection i and the waiting queue length on the road segments entering intersection i.

Action *A*: $A_t = \{a_{i,t}\}_{i=1}^{N}$ is the joint action set of all agents at time t. In this work, each intersection agent selects phase p from the phase set as its action $a_{i,t}$ at time t.

State transition probability *P*: $P : S \times A_1 \times \cdot \cdot \times A_N \to S$ denotes the state transition function. Given the system state S_t and corresponding joint actions A_t of agents at time t, the system arrives at the next state S_{t+1} according to the state transition probability.

Reward *R*: $R_t = \{r_{i,t}\}_{i=1}^{N}$ is the joint reward set of all the traffic lights at time t, where $r_{i,t}$ is the immediate reward for agent i at time t, and we define $E = [\sum_{t=1}^{T} \gamma^{t-1} r_{i,t}]$ as the expected future reward of traffic agent i, where γ is the discount factor. In this work, $r_{i,t} = \sum_{l=1}^{l_i} -q_l$, where l_i is the number of entering lanes connected to intersection i and q_l is the queue length of lane l.

On the basis of the above, the problem addressed in the paper is that given the traffic network directed graph G, as well as the immediate reward $r_{i,t}$ for action $a_{i,t}$ made by traffic light agent i at time t, our aim is to select $a_{i,t}$ for traffic light agent i so that the global reward $\sum_{i=1}^{N} r_{i,t}$ will be maximized.

3. NCCLight Model

3.1 Overview

The overall model of NCCLight is shown in Fig. 2. Specifically, the model constructs a graph composed of intersection agents, then uses four blocks to learn the initial input information on the graph, and realizes information exchange in the neighborhood and cooperation between neighboring intersections. Figure 2 shows the internal structure of the four blocks as follows.

(1) An observation embedding module is used to obtain an information vector representation of intersection nodes.

(2) A graph convolution network (GCN) module is used for information propagation in the neighborhood and information feature extraction of intersection nodes.

(3) An NCC module introduces the NCC theory to realize cooperation between multiple agents.

(4) A decision module is used to estimate the state value, which is helpful for updating the agent policy.

Along this line, NCCLight uses the interaction between an agent and neighbor traffic light agents to form a consistent neighborhood cognition, solve the problem of observability, and alleviate the problem caused by a non-stationary environment, which is conducive to multi-intersection traffic light control at the system level. In the following sections, we will describe these four modules in more detail.

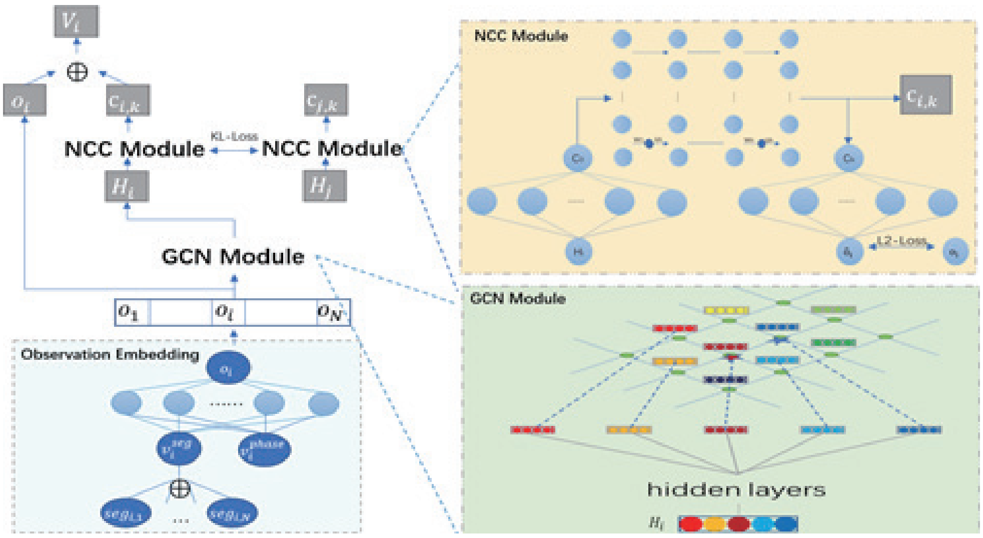


Fig. 2. (Color online) Overall model of NCCLight.

3.2 Independent advantage actor critic

Firstly, the MARL model based on IA2C is briefly introduced. In this paper, agent i's policy π_{θ_i} is a mapping from observation oi to action ai. Following policy π_{θ_i}, value function $V_{\pi_{\theta_i}}$ is defined as the expected discounted cumulative reward from observation o_i. Mathematically, it is given in a recursive form, namely, $V_{\pi_{\theta_i}}(o_{i,t}) = E[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}r_\tau \mid o_{i,t} = o]$, where γ is the discount factor. The value function $V_{\pi_{\theta_i}}$ captures the optimal expected cumulative reward that agent *i* can earn from observation o_i, and the state-action value $Q_{\pi_{\theta_i}}$ forces the agent to take action *a* and therefore captures the optimal expected cumulative reward from observation o_i and action a_i. Mathematically, $V_{\pi_{\theta_i}}(o_i) = \max_a Q_{\pi_{\theta_i}}(o_{i,t}, a_{i,t})$. In IA2C, the state value function $V_{\pi_{\theta_i}}$ is estimated by the w-parameterized critic, and then actor policy π_{θ_i} updates its parameter θ according to the direction suggested by the critic. Each update of θ increases the likelihood of selecting the "best" action. Figure 3 shows our IA2C implementation in the multi-agent scenario in detail. All agents have the same structure, and each agent independently learns its own strategy and corresponding value function. IA2C selects the state value function only based on the state as the benchmark function and obtains the advantage function, which is expressed as

$$A_{\pi_{\theta_i}}(o_{i,t}, a_{t,i}) = Q_{\pi_{\theta_i}}(o_{i,t}, a_{t,i}) - V_{\pi_{\theta_i}}(o_{i,t}). \tag{1}$$

Intuitively, the advantage function indicates the dominance of action $a_{t,i}$ in state $o_{i,t}$. When the value function of action $a_{t,i}$ is higher than the average value function, the item is positive, otherwise it is negative.

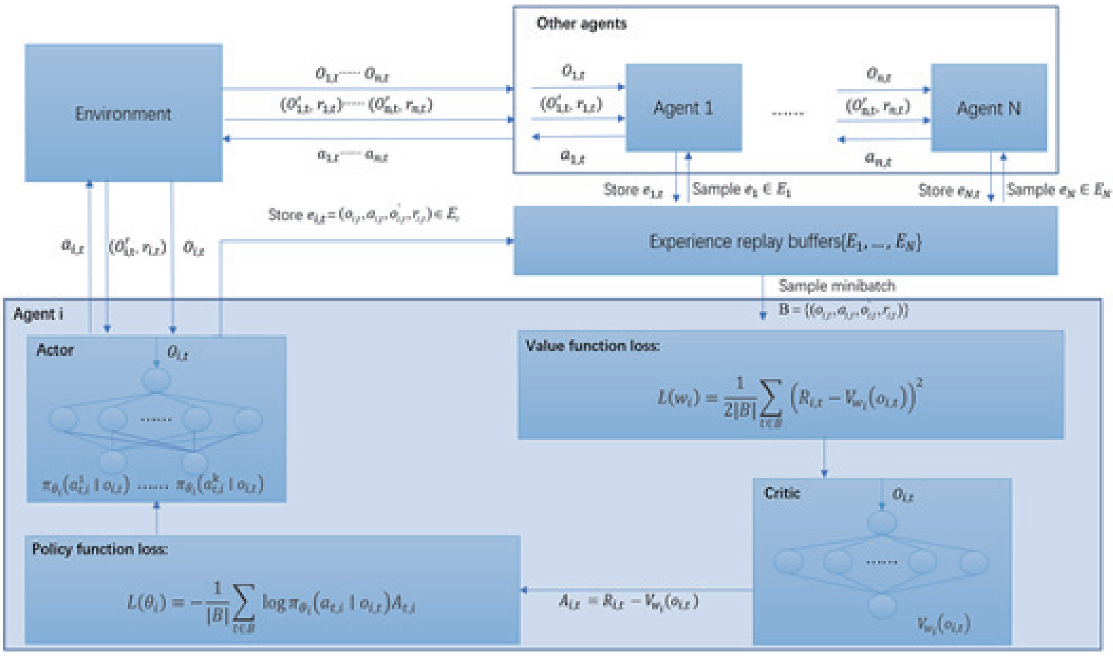

Other agents
Environment
$o_{1,t} \cdots\cdots o_{n,t}$
$(o'_{1,t}, r_{1,t}) \cdots\cdots (o'_{n,t}, r_{n,t})$
$a_{1,t} \cdots\cdots a_{n,t}$
$o_{1,t}$
$(o'_{1,t}, r_{1,t})$
$a_{1,t}$
Agent 1
......
$o_{n,t}$
$(o'_{n,t}, r_{n,t})$
$a_{n,t}$
Agent N
Store $e_{1,t}$
Sample $e_1 \in E_1$
Store $e_{N,t}$
Sample $e_N \in E_N$
Store $e_{i,t} = (o_{i,t}, a_{i,t}, o'_{i,t}, r_{i,t}) \in E_i$
Experience replay buffers$\{E_1, \ldots, E_N\}$
$a_{i,t}$
$(o'_{i,t}, r_{i,t})$
$o_{i,t}$
Sample minibatch
$B = \{(o_{i,t}, a_{i,t}, o'_{i,t}, r_{i,t})\}$
Agent i
Actor
$o_{i,t}$
......
$\pi_{\theta_i}(a^1_{t,i} \mid o_{i,t}) \cdots\cdots \pi_{\theta_i}(a^k_{t,i} \mid o_{i,t})$
Value function loss:
$L(w_i) = \frac{1}{2|B|}\sum_{t\in B}\left(R_{i,t} - V_{w_i}(o_{i,t})\right)^2$
Critic
$o_{i,t}$
......
$V_{w_i}(o_{i,t})$
Policy function loss:
$L(\theta_i) = -\frac{1}{|B|}\sum_{t\in B}\log\pi_{\theta_i}(a_{t,i} \mid o_{i,t})A_{t,i}$
$A_{i,t} = R_{i,t} - V_{w_i}(o_{i,t})$

Fig. 3. (Color online) Model of IA2C.

During training, each agent collects a minibatch $B = \{(o_{i,t}, a_{t,i}, o_{i,t+1}, r_t)\}$, which contains the experience trajectory, and each return is estimated as $\hat{R}_{t,i} = \sum_{\tau=t}^{t_B} \gamma^{\tau-t} r_\tau$, where t_B is the last step in the minibatch. A state value $R_{t,i} = \hat{R}_{t,i} + \gamma^{t_B-t} V_{w_i} -(s_t)$ is added to reduce the deviation of the sampling revenue, and $A_{\pi_{\theta_i}}(s_t, o_{i,t}) = R_{t,i} - V_{\pi_{\theta_i}}(s_t)$ is used instead of Eq. (1) to reduce the variance of the sampling revenue. In this case, the policy loss function can be expressed as

$$L(\theta_i) = -\frac{1}{|B|} \sum_{t \in B} \log \pi_{\theta_i}(a_{t,i} | s_t) A_{t,i}. \tag{2}$$

The loss function of value updating is

$$L(w_i) = \frac{1}{2|B|} \sum_{t \in B} \left(R_{t,i} - V_{w_i}(s_t) \right)^2. \tag{3}$$

For IA2C, the network parameters are shared among agents, which can reduce the number of network parameters to be proportional to the number of traffic light agents. In other words, the first encoder layer separately processes heterogeneous input information and shares the parameters of other layers.

3.3 NCCLight

Next, we introduce the learning process of the NCCLight model, which considers the neighborhood information of traffic signal agents to make agents coordinate better. IA2C is the most direct way to extend A2C to a multi-agent environment. There is no clear communication and cooperation between agents, and each agent's strategy only depends on the local state and local action. However, the NCCLight model will make the agent be affected by partial observability and a non-static MDP. This is because agents implicitly treat the action of other agents as part of the dynamic environment, and the strategies of other agents are constantly updated during training. In the field of ATSC, the behaviors of traffic light agents interact with each other, particularly when the intersections are tightly coupled. Therefore, the communication and cooperation among multiple agents need to be considered in the modeling process. Owing to the communication delay, global information sharing cannot be realized in real-time ATSC, so a model is needed to make use of the neighborhood information of these traffic lights to better communicate and cooperate, and thus optimize global traffic conditions.

3.3.1 Observation embedding module

In this problem, the original input information observed by the traffic light agent is the road segment information (the waiting queue length of each road segment entering the intersection) and the intersection node information (the current phase of the intersection signal).

The traffic information collected for each road segment is $seg_{ij} = \{\{q_l\}_{l=1}^{l_k}\}$, where q_l is the waiting queue length of the lth lane in the road segment and l_k is the number of lanes in the road segment. To transform the original input information into the embedded observation vector seg'_{ij}, each road segment uses the road segment encoder f_{seg_k} to encode the collected information seg'_{ij} calculated as

$$seg'_{ij} = f_{seg_{ij}}(seg_{ij}), \tag{4}$$

where $f_{seg_{ij}}$ is a two-layer multilayer perceptron (MLP) with the ReLU activation function.

To deal with heterogeneous information in the real world and reduce the number of road segment encoders with different dimensions, the first layer of the road segment encoder uses separate parameters to encode the information of different dimensions, and the parameters of other layers are shared.

To obtain the road segment information vector v_i^{seg} of intersection i, it is necessary to sum the observation vectors of the road segments entering the intersection as

$$v_i^{seg} = \sum_{j=1}^{N} seg'_{ij}, \tag{5}$$

where the number of road segments entering intersection i is N.

Then, the information vector of intersection i is connected with the observed intersection node information (phase $v_i^{phase} = \{Phase\, i\}$ of the intersection signal), and the information vector representation of intersection node i is obtained. The specific update formula is

$$o_i = f_{o_i}(v_i^{seg}, v_i^{phase}), \tag{6}$$

where f_v is a one-layer MLP.

3.3.2 GCN module

It can be seen from the above that the information vector of an intersection node only contains the local information observed by the intersection agent. Therefore, to model the overall influence of the neighborhood on the intersection agent, it is necessary to combine the information vectors of intersection nodes in the neighborhood with its own information vector. Herein, we use a GCN module to aggregate the information vectors of all intersections in a neighborhood to achieve information dissemination and feature extraction, and further extract higher-level information feature vectors H_i through the following equation:

$$H_i = \sigma\left(W \Sigma_{j \in N(i) \cap \{i\}} \frac{o_j}{C_{N(i)}} \right), \tag{7}$$

where $C_{N(i)}$ is the normalization constant. Symmetric normalization $C_{N(i)} = \sqrt{|N(j)||N(i)|}$ is chosen in this paper because it can reduce the weights of high-order neighborhoods more adaptively. As can be seen from Eq. (7), an idea similar to the mean field is used in this paper to sum the information vector representations of all intersections in the neighborhood; therefore, it is not necessary for all intersection nodes to align their indexes in the neighborhood. All intersection agents utilize the same parameter W to generate H_i. This parameter-sharing strategy based on the mean field can model adjacent intersections without an index, which can greatly reduce the total number of parameters in the learning model, reduce the risk of overfitting, and increase the robustness of the model.

3.3.3 NCC module

In the GCN module, the agent only interacts with the neighbor agents with local perception, so the problem caused by a non-stationary environment still exists. To resolve this problem, the NCC theory of social psychology is introduced to realize cooperation between agents. In a large-scale environment, neighbors have similar perception and are closely related; hence, they tend to form a consistent cognition of their own neighborhood environment. This inspired the idea that if neighboring agents have a consistent cognition of the environment in multi-agent cooperation, they can achieve system-level cooperation. For example, in traffic signal light control, an intersection may be unblocked but a neighboring intersection may be in a state of congestion. When neighboring agents have a consistent cognition of the traffic situation, they can work together to alleviate the traffic situation in their neighborhood, rather than just focusing on solving their own local traffic congestion.

In the NCC module, a high-level vector H_i containing neighbor information is used to learn the neighborhood cognitive variable C_i of intersection agent i, where C_i represents the cognition of intersection agent i to the general information of traffic congestion in the neighborhood. If neighboring agents can obtain the true hidden neighborhood cognitive variable C, they will eventually form consistent neighborhood cognition and achieve better cooperation. In other words, the neighborhood state cognitive variable c_k acquired by each agent should be similar to the true hidden cognitive variable C.

We start with the assumption that there exists a hidden process $p(o_i|C)$ for each agent, and we aim to infer C by evaluating $p(C|o_i)$. However, directly computing $p(C|o_i)$ is very difficult; hence, we approximate it by another distribution $q(c_{i,k}|o_i)$ that we introduce on the basis of stochastic gradient variational inference. Figure 2 illustrates the NCCLight model obtained by establishing flexible posterior distributions through an iterative procedure. The inference model builds a mapping from the observations o_i to $c_{i,0}$ with the parameters of the initial density $q_0(c_{i,0}\,|\,o_i) = N(\mu_0, \sigma_0)$, the output μ_0, and σ_0. We extract a random sample $\varepsilon \sim N(0,I)$ and initialize the chain with $c_{i,0} = \mu_0 + \sigma_0 \odot \varepsilon$.

F is constructed to denote the chain of composite functions: $F = f_K \circ f_{K-1} \cdots \circ f_2 \circ f_1$. In this way, the relationship between the data c_0 and the latent variable c_k can be represented as $c_{i,0} \overset{f_1}{\leftrightarrow} c_{i,1} \overset{f_2}{\leftrightarrow} c_{i,2} \ldots \overset{f_k}{\leftrightarrow} c_{i,k}$. In particular, we consider a family of such invertible transformations with a known Jacobian determinant, namely,

$$f_t(c_{i,t}) = c_{i,t-1} + u_t h_t(w_t^T c_{i,t-1} + b), \tag{8}$$

where u_t and w_t are vectors, w_t^T is the transpose of w_t, b is a scalar, and $h_t(\cdot)$ is a nonlinear function with derivative $h_t'(\cdot)$, such that $u_t h_t(w_t^T c_{t-1} + b)$ can be interpreted as an MLP with a bottleneck hidden layer with a single unit. We can compute the probability density function of the last iteration as

$$\log q_k(c_{i,k} \mid o_i) = \log q_0(c_{i,0} \mid o_i) + \sum_{t=1}^{K} \log \left| \det(\frac{dc_{i,t}}{dc_{i,t-1}}) \right|. \tag{9}$$

The generative model learns a mapping from c_k back to $\hat{o}_i$. In practice, the neighboring agents' cognitive distribution $p(c_{i,k}|o_i;w_i)$ is used to approximate the cognitive-dissonance loss. During numerical optimization, we need to iteratively minimize the following:

$$\psi_t(z_{t-1}) = h_t'(w_t^T z_{t-1} + b)w_t\,, \tag{10}$$

$$\min L2(o_i, \hat{o}_i; w) + \frac{1}{|N(i)|} \sum_{j \in N(i)} KL\Big(q(c_{i,k} o_{i,t}; w) \,\|\, q(c_{j,k} o_{j,t}; w)\Big) - \sum_{t=1}^{K} \ln |1 + \boldsymbol{u}_t^\top \psi_t(z_{t-1})|\,. \tag{11}$$

In the NCC module, all neighboring agents will eventually achieve consistent cognition at the neighborhood level by learning a cognitive variable c_k that is aligned well with the true hidden cognitive variable C.

3.3.4 Decision module

Ultimately, the distributed decision of each intersection agent can be obtained using the information vector representation of the intersection node. The decision of each agent i at time t is calculated as $\pi_{\theta_i} = f_{\theta_i}(o_{i,t})$, where f_{θ_i} is a two-layer MLP with ReLU activation.

On the basis of the above-mentioned cognitive variables of the neighborhood, every intersection agent calculates the value function V_w at every time step t. As shown in Fig. 2, for these intersection agents, we adopt element-wise summation to aggregate $o_{i,t}$ and $c_{i,k}$. The value function of each agent i at time t is calculated as

$$V_{w_i} = f_{w_i}(o_{i,t} \oplus c_{i,k}), \tag{12}$$

where f_{w_i} is also a two-layer MLP with ReLU activation.

3.4 Training

In addition, the technical solution of the training process is briefly introduced. Formally, agent i observes state $o_{i,t}$ at time step t and chooses an action $a_{i,t}$ based on the policy network π_{θ_i}. After the execution of $a_{i,t}$, the agent can observe a new observation $o'_{i,t}$ and obtain a reward $r_{i,t}$. Then, an experience tuple $e = (o_t, a_t, o'_t, r_t)$ is stored in the experience replay buffer D. Similarly, other agents in the environment store experience tuples independently in their experience replay buffers. During training, experience tuples are sampled from the agent's experience replay buffer and used to update the critic network by minimizing the combination of $L_i^{critic}(w)$ and $L_i^{VAE}(w)$. $L_i^{critic}(w)$ is the value network loss given by

$$L_i^{critic}(w) = \frac{1}{2|B|}\sum_{t\in B}\left(R_{t,i} - V_w(o_{i,t} \oplus c_{i,k})\right)^2 . \tag{13}$$

Each update of the parameters of the value network will allow the agent to better estimate the value function given the observation and action. $L_i^{NF}(w_i)$ is the normalizing flow network loss function.

$$L_i^{NF}(w) = \frac{1}{|B|}\sum_{t\in B}\left[L2(o_{i,t}, \hat{o}_{i,t}; w) + \frac{1}{|N(i)|}\sum_{j\in N(i)} KL\left(q(c_{i,k}o_{i,t}; w) \,||\, q(c_{j,k}o_{j,t}; w)\right) - \sum_{j\in N(i)} \ln|1 + \boldsymbol{u}_t^{\top}\psi_t(z_{t-1})|\right] \tag{14}$$

The total loss is a combination of $L_i^{critic}(w_i)$ and $L_i^{NF}(w_i)$ given as

$$L^{total}(w) = \sum_{i=1}^{N}[L_i^{critic}(w) + L_i^{NF}(w)], \tag{15}$$

where N is the number of intersections in the entire road network and w represents the trainable variables of the critic network in NCCLight.

The loss function of the actor network is

$$L(v) = -\frac{1}{|B|}\sum_{i=1}^{N}\sum_{t\in B}\log\pi_{\theta_i}(a_{i,t}\,|o_{i,t})A_{i,t}. \tag{15}$$

4. Experiments

In this section, we evaluate the effectiveness of the proposed model in multi-intersection traffic light control.

4.1 Experimental settings

CityFlow is an open-source simulator that can reproduce large-scale urban traffic scenes. It can not only show the real-world road network and traffic flow, but also provide application programming interfaces for MARL. This is conducive to the modeling of traffic signal control and other tasks. Therefore, the simulator is widely used by researchers.

In the experiment, we employed CityFlow to simulate traffic movements. First, the real-time traffic data is input into the simulator, and then the vehicle moves towards to its destination taking into consideration the environmental settings. The simulator supplies the status to the signal control model and further performs the traffic signal actions. According to the prescriptive set, each green signal is followed by a yellow signal of 3 s and a red signal of 2 s.

4.1.1 Synthetic datasets

The details are as follows.

Grid$_{6\times6}$-Uni: a 6 × 6 grid network with unidirectional traffic from west to east and from south to north. On the basis of practical experience, the traffic flow is generated using a Bernoulli distribution with probability 0.2, and the maximum number of vehicles arriving at an intersection is limited to three per second to ensure a stable simulation.

Grid$_{6\times6}$-Bi: a 6 × 6 grid network with bidirectional traffic in both the west–east and south–north directions. On the basis of objective facts, this traffic flow is generated using a Bernoulli distribution with probability 0.1, and we stabilize the simulation by setting the maximum number of vehicles arriving at an intersection to four per second.

4.1.2 Real datasets

Three real datasets are used here.

D$_{\textbf{New York}}$: an open-source taxi trip dataset of New York’s Upper East Side. There are 196 intersections in this traffic network.

D$_{\textbf{Hangzhou}}$: a publicly available dataset of Hangzhou city, China. The road structure in this dataset is a 4 × 4 grid and the traffic flow duration is 1 h.

D$_{\textbf{Jinan}}$: a publicly available real-world dataset of Hefei city, China. The traffic dataset is collected near 12 intersections in the Dongfeng subdistrict of Jinan.

4.2 Criterion

In our experiment, the commonly used average driving time (in s) of all vehicles entering and leaving the area is used as the criterion to evaluate the performance of different models. The average driving time, denoted as avgt, is calculated as

$$avgt = \frac{1}{N_c}\sum_{i=1}^{N_c} t_{i,ar} - t_{i,le}, \tag{17}$$

where N_c is the total number of vehicles entering an intersection, and $t_{i,ar}$ and $t_{i,le}$ are the times when the ith vehicle arrives at and leaves the intersection, respectively.

4.3 Benchmarks

To verify the effectiveness of NCCLight, both traditional models and several state-of-the-art models are chosen as benchmarks.

4.3.1 Traditional models

Fixed-time:(21) this method uses a predefined traffic lights control plan.
Max Pressure:(22) this method "greedily" selects the phase that maximizes the pressure to optimize the network-level traffic light control.

4.3.2 State-of-the-art models

CGRL:(23) a multi-intersection signal control model based on reinforcement learning (RL), in which agents are trained to optimize the joint actions between intersections. This approach decreases the average travel time compared with some earlier models based on RL for traffic light control.
Individual RL:(24) an RL model in which agents maintain their own parameters independently without considering the influence of neighbors. The observation of an agent is the local traffic state of intersections. In some simple scenarios, this model performs well, allowing vehicles to pass quickly.
OneModel:(25) a model based on an individual RL model, in which all agents share the same policy network. This model has comparable performance with the optimized fixed-time strategy and also with the vehicle-driven controller, and the best results are obtained under a relatively high traffic load.
Neighbor RL:(26) an RL model based on the above OneModel. Specifically, agents concatenate their own observations with their neighborhood traffic conditions, and all agents share the same parameters. This model allows intersections to execute different decision strategies as long as they abide by the predetermined protocol rules.
GCN:(27) a traffic signal control model based on RL that uses the graph convolutional neural network to automatically extract the traffic characteristics of adjacent intersections. The algorithm minimizes the driving time on the road and improves the safety of residential areas.
CoLight:(28) a model that controls traffic lights by employing graph attention networks. This model determines the neighbors of an agent by predefined rules and defines the number of neighbors of each agent in advance. CoLight was the first model to apply the graph attention network to the RL algorithm for traffic signal control and has achieved superior performance in experiments on large-scale road networks with hundreds of traffic signals.

4.4 Performance comparison

In this section, we evaluate the performance of NCCLight by comparing it with the benchmarks introduced above and analyze the results of such evaluation.

4.4.1 Overall analysis

Table 1 shows the performance of NCCLight and the benchmarks on both synthetic and real traffic datasets. It is clear that NCCLight outperforms the other benchmarks. We also point out the following interesting observations:

When the evaluation data change from synthetic traffic data to a real-world dynamic traffic flow, some RL-based models, such as Individual RL and OneModel, have lower performance than the traditional traffic models, such as MaxPressure. The main reason is that the road structure is more irregular and the traffic flow is more dynamic in the real world than for synthetic data. Individual RL and OneModel are relatively crude in their design and ignore the information of neighbors; thus, their performance is unsatisfactory. By contrast, the performance of cooperative MARL models, such as Neighbor RL and GCN, is much better than that of the traditional traffic models. This demonstrates the effectiveness of cooperative MARL models, which take neighbors' information into consideration and adaptively change the traffic phase to optimize the long-term traffic conditions.

Also from Table 1, the performance of Neighbor RL and GCN is inferior to that of NCCLight. These two models perform well in the synthetic traffic flow, but not in a large-scale real traffic flow. It is found that in addition to its own traffic conditions, the intersection agent in these two models only considers the traffic conditions of its adjacent intersections according to static prior geographical knowledge. However, the congestion conditions of intersections are updated dynamically. For example, during the morning rush hour, a large number of vehicles drive from intersection *i* to intersection *j*, leading to traffic congestion in this direction. However, in the evening peak hours, the direction of congested sections may be opposite. Therefore, the inability to dynamically adjust their decision-making strategies results in the poor performance of these two models. In contrast, the agent in NCCLight dynamically updates its cognition of the traffic congestion in its neighborhood in real time, rather than on the basis of static prior knowledge.

Table 1
Performance comparison on synthetic data and real-world data in terms of average travel time in an area.

Model	**Grid$_{6\times6}$-Uni**	**Grid$_{6\times6}$-Bi**	**D$_{\text{New York}}$**	**D$_{\text{Hangzhou}}$**	**D$_{\text{Jinan}}$**
Fixed-time	241.73	217.68	2028.71	754 .37	899.34
MaxPressure	210.73	203.86	1733.64	565.34	504.2
CGRL	1672.84	2012.45	2269.10	1679.86	1320.90
Individual RL	307.41	281.42	4169.74	667.54	595.23
OneModel	179.30	257.98	2058.91	537.29	445.63
Neighbor RL	233.12	237.15	1672.92	443.56	485.13
GCN	241.22	284.01	1538.39	563.29	457.53
CoLight	181.96	198.16	1432.85	297.14	284.7
NCCLight	176.23	169.42	1329.17	241.43	238.72

Therefore, when traffic congestion occurs, agents can adjust their decision-making strategies according to the consistent cognition of traffic congestion of all the neighborhood agents.

From Table 1, NCCLight is superior to CGRL. To coordinate multiple agents, CGRL factorizes the global Q-function as a linear combination of local subproblems to optimize the joint global action over the entire coordination graph. However, the size of the joint action space is directly proportional to the number of intersections, and this causes a scalability problem. In contrast, NCCLight adopts a distributed structure where each agent learns its own policy and the corresponding value function. During the training process, NCCLight adopts a way of parameter sharing that reduces the number of parameters to be trained. Therefore, NCCLight achieves better performance than CGRL.

Moreover, it can be seen from Table 1 that NCCLight outperforms CoLight on all the datasets. The agent in CoLight integrates the temporal and spatial effects of its adjacent intersection agents by employing a graph neural network. Even though the agent in NCCLight also adopts a graph neural network to connect the observation vector representation of its neighbors, the agent in NCCLight not only expands the observation range but also has a high-level cognition of the traffic congestion in the whole neighborhood area. In the real world, cognitive consistency has a critical role in maintaining human social order. Therefore, we employ a normalizing flow to construct a reversible transformation model, so that the initial observation vector is transformed into a high-level feature vector representing the neighborhood cognition, and we add the constraint of cognitive consistency to make the agents in the neighborhood agree on the cognition of traffic congestion. As can be seen from Table 1, our model achieved the best results, which demonstrates that it is effective to adopt neighborhood consistency theory to model multiple cooperative agents.

4.4.2 Comparison of convergence

A comparison of the convergence of the traffic models based on RL is illustrated in Fig. 4. This figure shows the learning curves of NCCLight and five other RL models in the 100-episode training process for all the datasets. NCCLight exhibits excellent initial performance in the first episode, quickly achieves the predetermined performance, and ends with the optimal strategy. The results clearly show that learning the consensus of multi-agent cognition in the neighborhood does not reduce the convergence speed of the model, whereas it increases the speed of convergence to the optimal strategy.

4.4.3 Scalability comparison

Here, we demonstrate that NCCLight is more flexible than the other RL-based traffic models in terms of performance, training time, and the number of neighborhood nodes.

- **Performance**

From the convergence curves in Fig. 4 and the average travel times in Table 1, it can be observed that NCCLight performs better than the other RL models in large real networks. Its performance is best even in a real-world network of 196 intersections.

(f)

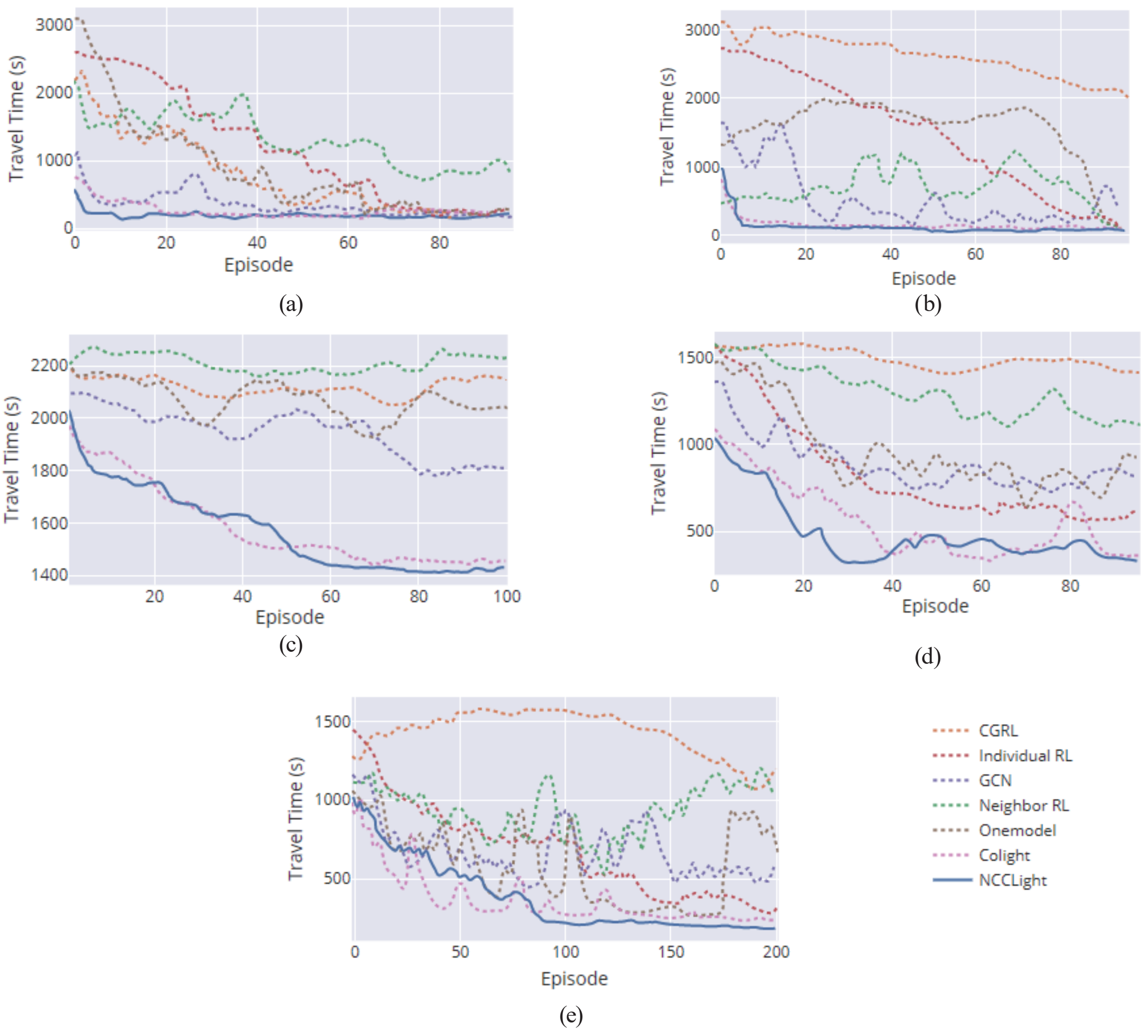


Fig. 4. (Color online) Learning curves of the NCCLight model and five other RL models on five datasets. Curves are smoothed by taking the moving average of five points. (a) $\mathbf{Grid_{6\times6}}$**-Uni**, (b) $\mathbf{Grid_{6\times6}}$**-Bi**, (c) $\mathbf{D_{New\ York}}$, (d) $\mathbf{D_{Hangzhou}}$, and (e) $\mathbf{D_{Jinan}}$.

- **Training time**

The 100-episode training time of NCCLight on road networks of different scales is compared with those of the other five RL models. All models are evaluated separately on the same server to ensure a fair comparison. As illustrated in Fig. 5, the training time of NCCLight is equivalent to those of OneModel, GCN, and CoLight, and much less than those of CGRL, Individual RL, and Neighbor RL. The reasonable explanation for this is that NCCLight adopts the strategy of sharing parameters without an index, which greatly reduces the training time. Therefore, in addition to providing high performance, the training efficiency of NCCLight is also reasonably high.

- **Effect of number of neighbors**

Figure 6 shows the effect of the number of neighbors on the performance of NCCLight. With increasing number of neighbors, the performance of NCCLight improves and tends to converge to the optimal value. In particular, when there are four neighbors, the performance of NCCLight

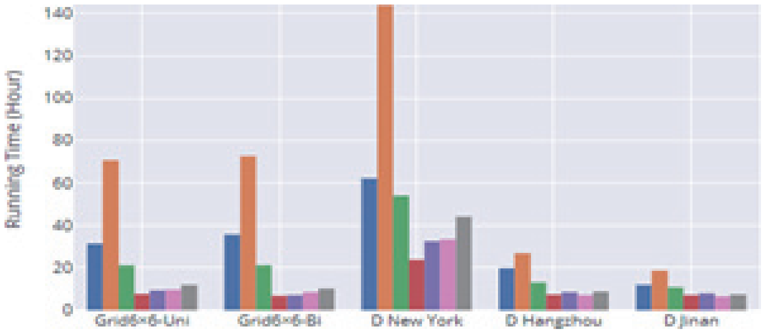
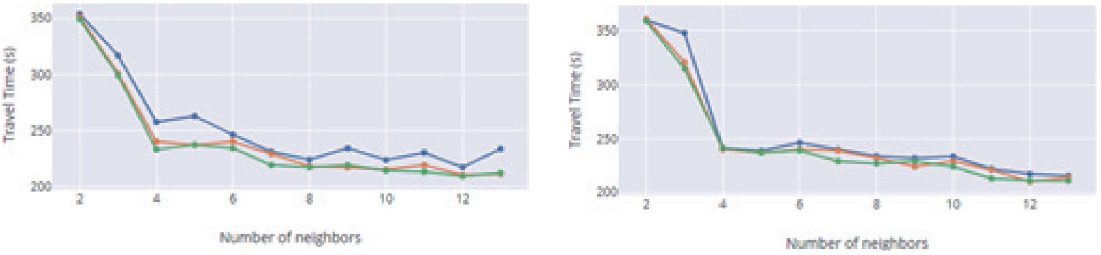


Fig. 5. (Color online) Training times of 100 episodes with different RL models. NCCLight's training is efficient across all datasets.



(a)

(b)

Fig. 6. (Color online) Performance of NCCLight with respect to number of neighbors (|Ni |) on (a) DHangzhou and (b) DJinan datasets.

is close to optimal. This is because the intersection has the closest relationship with its neighbors in this case. When the information of distant neighbors is added, the training performance of NCCLight is not greatly improved, but the training time is increased. Therefore, when determining the signal control strategy of each intersection, the observation results of only four adjacent intersections should be considered, which can ensure both efficient training and high performance.

5. Conclusions

We successfully constructed a scalable and fully decentralized IA2C model based on NCC with the aim of improving MARL in adaptive traffic signal control. It possesses the following innovations: 1) On the basis of the constructed traffic graph, an index-free model with parameter sharing is constructed by leveraging a graph convolution network, which realizes information exchange between intersections. 2) A normalizing flow is introduced so that all neighborhood agents can learn neighborhood cognition together to obtain the same neighborhood cognition and thus achieve better cooperation. Both real and synthetic data were used to demonstrate the robustness, optimality, and scalability of the designed NCCLight model, which outperforms other advanced adaptive traffic signal control models based on MARL.

Furthermore, this promising design strategy based on MARL provides a blueprint for application to various complex and varied environments. Weather conditions should be taken into account to further increase the accuracy of the model, such as unsettled weather during the rainy season, i.e., specific extra data must be included in the model under certain circumstances. To enable an effective response to various types of traffic flow information, the number of neighbors in the neighborhood should also be determined in a more flexible way.

References

1 A. J. Miller: J. Oper. Res. Soc. **14** (1963) 373. https://doi.org/10.1057/jors.1963.61

2 S.-B. Cools, C. Gershenson, and B. D'Hoogh: Self-organizing Traffic Lights: A Realistic Simulation, M. Prokopenko, Ed. (Springer, London, 2007) Chap. 3, pp. 41–49.

3 F. Rasheed, K. Yau, R. Md. Noor, C. Wu, and Y. Low: IEEE Access **8** (2020) 208016. https://doi.org/10.1109/ACCESS.2020.3034141

4 S. El-Tantawy, B. Abdulhai, and H. Abdelgawad: IEEE Trans. Intell. Transport. Syst. **14** (2013) 1140. https://ieeexplore.ieee.org/document/6502719

5 L. A. Prashanth and S. Bhatnagar: IEEE Trans. Intell. Transport. Syst. **12** (2011) 412. https://ieeexplore.ieee.org/document/5658157

6 L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis: Lect. Notes. Comput. Sci. **5211** (2008) 656. https://doi.org/10.1007/978-3-540-87479-9_61

7 T. T. Nguyen, N. D. Nguyen, and S. Nahavandi: IEEE Trans. Cybern. **50** (2020) 3826. https://ieeexplore.ieee.org/document/9043893

8 T. Chu, J. Wang, L. Codecà, and Z. Li: IEEE Trans. Intell. Transport. Syst. **21** (2020) 1086. https://ieeexplore.ieee.org/document/8667868

9 K. M. Dresner and P. Stone: 1st Int. Workshop on Learning and Adaption in Multi-Agent Systems (2006) 129–138.

10 L. Buşoniu, R. Babuška, and B.D. Schutter: Innovations in Multi-Agent Systems and Applications – 1. Studies in Computational Intelligence (Springer, Berlin, 2010) Vol. 310, pp. 183–221.

11 H. Mao, W. Liu, J. Hao, J. Luo, D. Li, Z. Zhang, J. Wang, and Z. Xiao: 34th AAAI Conf. Artificial Intelligence (AAAI, 2019) 7219–7226.

12 I. Arel, C. Liu, T. Urbanik, and A.G. Kohls: IET Intell. Transp Syst. **4** (2010) 128. https://doi.org/10.1049/iet-its.2009.0070

13 M. A. Khamis and E. Gomaa: 11th IEEE Int. Conf. Machine Learning and Applications (ICMLA, 2012) 586–591.

14 M. Norouzi, M. Abdoos, and A. L. C. Bazzan: J. Supercomput. **77** (2021) 780. https://doi.org/10.1007/s11227-020-03287-x

15 T. Nishi, K. Otaki, K. Hayakawa, and T. Yoshimura: 21st IEEE Int. Conf. Intelligent Transportation Systems (ITSC, 2018) 877–883.

16 D. Simon, C. J. Snow, and S. J. Read: J. Pers. Soc. Psychol. **86** (2004) 814. https://doi.org/10.2139/ssrn.439984

17 J. E. Russo, K. A. Carlson, M. G. Meloy, and K. Yong: J. Exp. Psychol. Gen. **137** (2008) 456. https://doi.org/10.1037/a0012786

18 A. Bear, A. Kagan, and D. G. Rand: Proc. Biol. **284** (2017) 20162326. https://doi.org/10.1098/rspb.2016.2326

19 H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, Z. Li, and H. Zhang: Proc. 2019 World Wide Web Conf. (WWW, 2019) 3620–3624.

20 N. Brazil and D. S. Kirk: AM J. Epidemiol. **184** (2016) 192. https://doi.org/10.1093/aje/kww062

21 A. Bear, A. Kagan, and D. G. Rand: Proc. Royal Soc. B **284** (2017) 1851. https://doi.org/10.1098/rspb.2016.2326

22 T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. Wu: IEEE Trans. Veh. Technol. **69** (2020) 8243. https://doi.org/10.1109/TVT.2020.2997896

23 H. Wei, G. Zheng, H. Yao, and Z. Li: Proc. 24th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD, 2018) 2496–2505.

24 T. Chu, J. Wang, L. Codecà, and Z. Li: IEEE Trans. Intell. Transp. **21** (2020) 1086. https://ieeexplore.ieee.org/document/8667868

25 S. Rizzo, G. Vantini, and S. Chawla: 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD, 2019) 1654–1664.

26 H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li: 28th ACM Int. Conf. Information and Knowledge Management (CIKM, 2019) 1913–1922.

27 M. Pedro, U. Wasim, and H. Jack: Transport. Res. C-Emer. **110** (2019) 275. https://doi.org/10.1016/j.trc.2019.10.002

28 K. Dresner and P. Stone: 1st Int. Workshop on Learning and Adaption in Multi-Agent Systems (2005) 129–138.