MaCAR: Urban Traffic Light Control via Active Multi-agent Communication and Action Rectification

Zhengxu Yu[1*], **Shuxian Liang**[2,3*], **Long Wei**[1], **Zhongming Jin**[3], **Jianqiang Huang**[3], **Deng Cai**[1], **Xiaofei He**[1,4] and **Xian-Sheng Hua**[3]
[1]State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
[2]Computer Science Department, Zhejiang University, Hangzhou, China
[3]DAMO Academy, Alibaba Group, Hangzhou, China
[4]Fabu Inc., Hangzhou, China
yuzxfred@gmail.com, {shuxian.lsx, longwei}@zju.edu.cn,
{zhongming.jinzm, jianqiang.hjq}@alibaba-inc.com,
{dengcai, xiaofeihe}@cad.zju.edu.cn, huaxiansheng@gmail.com

Abstract

Urban traffic light control is an important and challenging real-world problem. By regarding intersections as agents, most of the Reinforcement Learning (RL) based methods generate actions of agents independently. They can cause action conflict and result in overflow or road resource waste in adjacent intersections. Recently, some collaborative methods have alleviated the above problems by extending the observable surroundings of agents, which can be considered as inactive cross-agent communication methods. However, when agents act synchronously in these works, the perceived action value is biased and the information exchanged is insufficient. In this work, we propose a novel Multi-agent Communication and Action Rectification (MaCAR) framework. It enables active communication between agents by considering the impact of synchronous actions of agents. MaCAR consists of two parts: (1) an active Communication Agent Network (CAN) involving a Message Propagation Graph Neural Network (MPGNN); (2) a Traffic Forecasting Network (TFN) which learns to predict the traffic after agents' synchronous actions and the corresponding action values. By using predicted information, we mitigate the action value bias during training to help rectify agents' future actions. In experiments, we show that our proposal can outperforms state-of-the-art methods on both synthetic and real-world datasets.

1 Introduction

Urban Traffic Light Control (TLC) is a critical and challenging real-world problem, which aims to maximize the traffic efficiency with limited urban road resource and avoid traffic conflict inside intersections. Finding an appropriate TLC approach can significantly mitigate traffic congestion and bring in significant economic, environmental and societal benefits [Wei *et al.*, 2018].

Most of the recently proposed reinforcement learning (RL) based methods [Wei *et al.*, 2018; Nishi *et al.*, 2018; Casas, 2017] were focusing on independently controlling intersections in the same road network. In these works, the observable surrounding of an agent is limited to itself, which may cause action conflict between agents and deviate action value from expectation. One example of this action conflict is increasing green-light time in directions suffering heavy traffic can significantly mitigate the traffic. However, it may cause severe congestion in adjacent regions if the adjacent agents cannot adjust themselves in time. Such congestion often occurs because traffic changes caused by agent action changes can be rapid and abrupt.

To overcome this problem, some collaborative optimization based works [Van der Pol and Oliehoek, 2016; Nishi *et al.*, 2018; Chu *et al.*, 2019; Wei *et al.*, 2019b] were proposed recently. These works mitigate the action conflict problem mainly by expanding the observable surroundings of agents. It can be seen as establishing an inactive communication mechanism between agents, in which agents can obtain not only the traffic state of themselves but also adjacent intersections. This mechanism is inactive because agents do not directly share decision information such as actions and agent states, but share noised responses of the road network.

There are two shortcomings of this inactive communication mechanism. First, agents cannot know each other's new actions before perceiving the traffic changes in this inactive communication mechanism. Hence, the perceived action values of agents are biased, since agents perceive traffic pattern changes always lag behind action change. Secondly, the information propagated is insufficient for collaborative optimization, because the shared information only includes the surrounding traffic states, but not includes important agent decision information such as historical actions.

In this work, we propose a novel Multi-agent Communication and Action Rectification (MaCAR) framework, which consists of two parts. The first part is a Communication Agent Network (CAN) with a Message Propagation Graph

*Equal Contribution

Neural Network (MPGNN) based active communication network. The purpose of CAN is to generate actions considering other agents' actions and states. By introducing the action information, agents can effectively learn the traffic pattern under the corresponding action, helping reduce the action space and improve model performance. The second part of MaCAR is a novel Traffic Forecasting Network (TFN). TFN forecast the traffic of the whole road network under agents' new actions, together with the corresponding action values. Using this predicted information, we further amend the action value of agents during training to mitigate action value deviation.

We carried out experiments on both synthetic and real-world datasets to evaluate the performance of MaCAR. Experimental results demonstrate that by introducing the active communication mechanism and taking advantage of traffic forecasting information, our proposal can achieve superior performance against the state-of-the-art methods.

We summarize the contributions of this work as follows:

1. We focus on the action conflict problem in previous independent RL methods, which can leads to severe congestion at adjacent intersections when multiple agents act synchronously.
2. In this work, we propose a novel Multi-agent Communication and Action Rectification (MaCAR) framework to control multiple agents collaboratively. MaCAR consists of two components: (1) a novel Message Propagation based Communication Agent Network (CAN), which generates coordinated actions via an active cross-agent communication mechanism; (2) a novel Traffic Forecasting Network (TFN) which helps to rectify action against action value bias.
3. In experiments, we demonstrate that our proposal can achieve state-of-the-art performance on both synthetic and real-world datasets by involving the active communication mechanism and take advantage of traffic forecasting information.

2 Related Works

2.1 Reinforcement Learning based Traffic Light Control

Independent Control v.s. Collaborative Control

Most recently proposed methods [Aslani *et al.*, 2017; Wei *et al.*, 2018; Zheng *et al.*, 2019; Chen *et al.*, 2020] were focusing on optimizing agent independently, in which an independent RL agent is compelled to control a certain intersection without noticing other agents' existing. However, as part of the whole road network, the impact of other agents cannot be completely blocked out in real-world applications.

To mitigate this problem, some collaborative control based methods were proposed recently [Lee *et al.*, 2019; Van der Pol and Oliehoek, 2016; Wei *et al.*, 2019b]. Wei *et al.* [2019b] proposed the CoLight, an independent TLC algorithm with inactive communication mechanism. Their inactive communication mechanism extended agent's receptive field to help make coordinated action. Agents do not share decision information directly in their work, but rather share the road network's responses to historical actions. Therefore, agents do not know the latest actions of other agents until it feels the traffic changes, resulting in a possible action conflict. Different from them, MaCAR is a multi-agent collaborative control based method with an active communication mechanism. In MaCAR, we use the traffic prediction information to actively rectify agents' future actions during training by alleviating action value bias caused by other agents' actions.

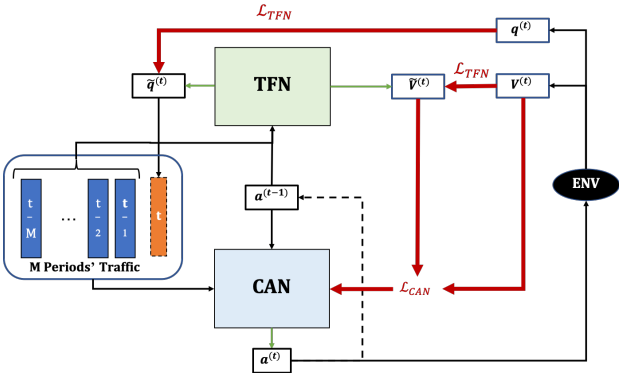


Figure 1: Framework of the Multi-agent Communication and Action Rectification (MaCAR).

Adaptive Control v.s. Pre-defined Phase Scheme

Most recently proposed RL methods [Wei *et al.*, 2018; Nishi *et al.*, 2018; Wei *et al.*, 2019a; Wei *et al.*, 2019b; Aslani *et al.*, 2017; Casas, 2017] were based on adaptive control strategy in which agent selects whether switch the permitted phase (lanes) per second based on traffic situation. Adaptive control provides excellent controllability since the permitted phase can be switched in seconds. However, frequent phase switching may cause traffic accidents and significantly affect the driving experience. To avoid bad driving experience and traffic accidents caused by a sudden phase switching, some pre-defined phase scheme based RL approaches [Aslani *et al.*, 2017; Casas, 2017] were proposed. A typical phase scheme consists of a fixed phase execution cycle and the corresponding execution time list. The agent's action is to adjust the phase execution time. Similarly, MaCAR using a pre-defined phase scheme based control strategy to provides practicability in real-world applications.

2.2 Graph Neural Network

Recently proposed GNNs can be divided into two categories, spectral-based (mainly GCN based) methods [Li *et al.*, 2018; Yu *et al.*, 2019], and Message Propagation based methods [Battaglia *et al.*, 2018; Wei *et al.*, 2019c]. A typical GCN based model that trained on a specific graph could not be directly applied to a graph with a different graph [Zhou *et al.*, 2018]. Hence, plain GCNs is inconvenient in a TLC scenario because multiple agents' actions can change the weight of each road tremendously. More recently, several Message Propagation based methods were proposed to solve this problem [Wei *et al.*, 2019c]. Similarly, MaCAR uses Message Propagation based neural networks to model dynamic graphs from traffic.

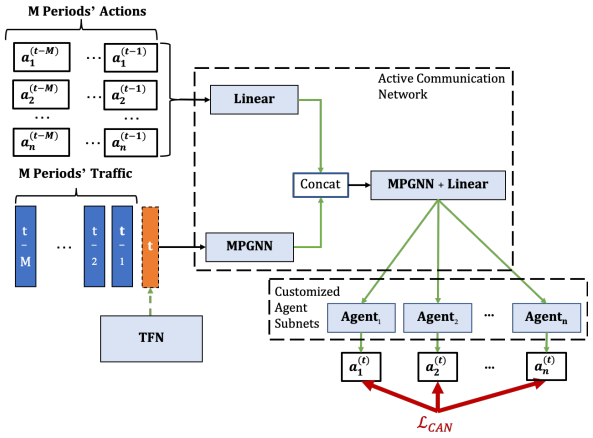


Figure 2: Sketch of the Communication Agent Network (CAN).

3 Our Method

We first introduce some notations and terms. We define term *period* as the time length of executing all phases once. For simplicity, we set the period length of all intersections as P to ensure agents' synchronous actions. The action applied in period t is defined as $\mathrm{a}^{(t)}$, and its corresponding perceived action value is defined as $\mathrm{v}^{(t)}$. The adjacent intersection set of intersection n is defined by $\mathcal{N}(n)$, and it is obtained from the road network topology G.

The input traffic state $\mathrm{q}^{(t)}$ is a tensor contains the queue lengths of all roads in the road network after t-th period. Our purpose is to minimize the queue length $\mathrm{q}^{(t)}$ of the whole road network, which is a common practice in previous works. To achieve that, we use a concatenated tensor contains the input traffic state and the corresponding actions of the past M periods as input, and we denote this concatenated tensor as $\{\{\mathrm{q},\mathrm{a}\}^{(t-\mathrm{M})},...,\{\mathrm{q},\mathrm{a}\}^{(t-1)}\}$, where M is a training interval hyper-parameter selected by practice.

To provide good agent customization capability in real-world applications and avoid causing bad driving experience and traffic accidents, we use pre-defined phase scheme based control strategy in MaCAR. A typical action of MaCAR is to redefine the intersection's phase execution time list at the beginning of a new period. The pre-defined phases can be hand-crafted by experts according to real-world applications.

3.1 Communication Agent Network

The framework of MaCAR has shown in Figure 1, the first part is the Communication Agent Network (CAN). CAN is a multi-agent action generation network consisting of an MPGNN based central active communication network and several customized agent subnets, as shown in Figure 2. In CAN, the embedding of the previous M periods' traffic states and its corresponded actions are extracted and propagated via the active communication network, and then feed into the customized agent subnets. By introducing the decision information, we impel the model to learn the message propagation pattern while considering the impact of agents' actions.

Active Communication Network

The active communication network of CAN is formed by several linear layers and Message Propagation based Graph Neural Networks (MPGNNs) which is designed by following the strong discriminant graph theory proved by Xu *et al.* [2019]. MPGNN learns the spatio-temporal message propagation mechanism between agents.

MPGNN has a multi-layered network architecture, which can learn complex propagation pattern with a wide receptive field. Formally, the feature embedding of intersection n in k-th layer $\mathrm{h}_n^{\mathrm{k}}$is:

$$\mathrm{m}_n^{\mathrm{k}} = f_p(\sum_{u\in\mathcal{N}(n)} \mathrm{h}_u^{\mathrm{k}-1}), \tag{1}$$

$$\mathrm{h}_n^{\mathrm{k}} = f_a((1+\epsilon^{\mathrm{k}})\mathrm{h}_n^{\mathrm{k}-1} + \mathrm{m}_n^{\mathrm{k}}), \tag{2}$$

where $\mathrm{m}_n^{\mathrm{k}}$ is the aggregated message received from adjacent intersections of intersection n in k-th layer. f_p and f_a is the propagation and aggregation function respectively, both of them are learned by using a neural network consists of two linear layers. ϵ^{k} is a learnable parameter which helps to generate discriminant graph representation. The input intersection features $\{\{\mathrm{q}_n,\mathrm{a}_n\}^{(t-\mathrm{M})},...,\{\mathrm{q}_n,\mathrm{a}_n\}^{(t-1)}\}$ are defined as h_n^0. The output of the last layer of MPGNN $\mathrm{h}^{\mathrm{out}}$ is a graph embedding contains all aggregated features $\{\mathrm{h}_n^{\mathrm{out}} \mid n \in G\}$.

Based on MPGNN, we establish the active communication network to learn the message aggregation and propagation among agents under certain actions, as shown in Figure 2. We then feed the output into agent subnets to generate actions.

Customized Agent Subnet

To fit different types of intersection's traffic light settings, agents in CAN are still decentralized agents. The intersection number determines the number of customized agent subnets. Meanwhile, agent subnet is customized according to the corresponding intersection settings, such as its channelization or direction number. By doing so, MaCAR can fit most existing intersection settings and commonly used control strategies.

Each agent (subnet) of CAN learns a continuous action distribution $\mathcal{P}$. We then sample action from the learned distribution using following equations:

$$Sample\ \mathrm{a}_n^{(t)} \sim \mathcal{P}_n(f_n(\mathrm{h}^{\mathrm{out}})), \tag{3}$$

$$\mathrm{a}_n^{(t)} = softmax(\alpha\ \mathrm{a}_n^{(t)}), \tag{4}$$

where n is the intersection index, f_n learns the mapping from the global feature embedding $\mathrm{h}^{\mathrm{out}}$ to the mean and variance of action distribution $\mathcal{P}_n$.

Action sampled from the continuous distribution $\mathcal{P}$ is first uniformed by using a uniform multiplier $\alpha > 0$, and then rescaled by using a Softmax function to generate the new phase executing time ratio $\mathrm{a}_n^{(t)}$. At last, we multiply $\mathrm{a}_n^{(t)}$ and P to get the new execution time list.

3.2 Traffic Forecasting Network

The second part of MaCAR is the Traffic Forecasting Network (TFN), which aims to predict the future traffic and the action value of the given actions. By using this forecasting

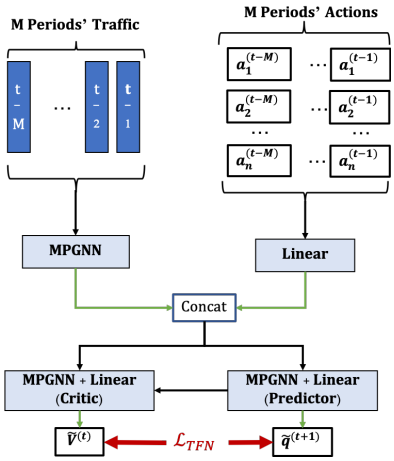


Figure 3: Sketch of the Traffic Forecasting Network (TFN).

information, we amend the action value and further rectify agent action against other agents' actions.

As shown in Figure 3, TFN first extracts feature embeddings from the previous M periods' traffic using an MPGNN (k=2), and then concatenates them with the embeddings of the corresponding actions. The concatenated embeddings are then fed into two branches (Predictor and Critic). The Predictor branch is used to predict the future traffic $\widetilde{\mathrm{q}}^{(t)}$ of the whole road network. The Critic branch aims to predict the action value $\widetilde{\mathrm{v}}^{(t)}$ of the given actions $\mathrm{a}^{(t)}$.

As shown in Figure 3, there is a shortcut from the Predictor branch to the Critic branch. We argue that this shortcut can bring in the predicted future traffic under the given actions to help mitigate the impact of other agents' actions on $\widetilde{\mathrm{v}}^{(t)}$.

3.3 Online Training Algorithm

We devise a policy gradient based online training algorithm to train the CAN and TFN after each specific periods M. The online training algorithm has shown in Algorithm 1.

As shown in Algorithm 1, to collect the initial information for MaCAR, we first let simulator run for M periods under initial action $\mathrm{a}^{(0)}$ in which all phases have the same executing time. After that, new predictions and actions are generated by using the traffic states and corresponding actions of previous M periods. We train the MaCAR network after every M periods, in which one period is equal to P time steps. During training, we first calculate the action values of the past M periods by using Eq. 5. The action values of t-th period are defined by the difference of queue lengths between $(t-1)$ and (t) period:

$$\mathrm{v}^{(t)} = \mathrm{q}^{(t-1)} - \mathrm{q}^{(t)}. \tag{5}$$

We then optimize the TFN network by minimizing the following loss function:

$$\mathcal{L}_{\mathrm{TFN}} = \sum_{t=0}^{\mathrm{M}} (||\mathrm{v}^{(t)} - \widetilde{\mathrm{v}}^{(t)}||_{\ell_1} + ||\mathrm{q}^{(t)} - \widetilde{\mathrm{q}}^{(t)}||_{\ell_1}), \tag{6}$$

where $||\cdot||_{\ell_1}$ represents ℓ_1 norm. It's worth noting that both $\widetilde{\mathrm{v}}^{(t)}$ and $\widetilde{\mathrm{q}}^{(t)}$ are generated before generating $\mathrm{a}^{(t)}$.

Algorithm 1 Online Training algorithm

Input: Initial action $\mathrm{a}^{(0)}$, parameters θ_{TFN} and θ_{CAN}, period length P, simulation time length t_{max}, training interval M, simulator $\mathcal{S}$
Output: Optimized θ_{TFN} and θ_{CAN}

Let $t = \mathrm{M}$
run $\mathcal{S}(\mathrm{a}^{(0)})$ M periods $\rightarrow \{\{\mathrm{q},\mathrm{a}\}^{(0)}, ..., \{\mathrm{q},\mathrm{a}\}^{(t-1)}\}$
$\mathrm{TFN}(\{\{\mathrm{q},\mathrm{a}\}^{(0)}, ..., \{\mathrm{q},\mathrm{a}\}^{(t-1)}\}; \theta_{\mathrm{TFN}}) \rightarrow \widetilde{\mathrm{v}}^{(t)}, \widetilde{\mathrm{q}}^{(t)}$
$\mathrm{CAN}(\{\{\mathrm{q},\mathrm{a}\}^{(0)}, ..., \{\mathrm{q},\mathrm{a}\}^{(t-1)}\}, \widetilde{\mathrm{q}}^{(t)}; \theta_{\mathrm{CAN}}) \rightarrow \mathrm{a}^{(t)}$
$\mathcal{S}(\mathrm{a}^{(t)}) \rightarrow \{\mathrm{q},\mathrm{a}\}^{(t)}$
$t = t + 1$
while $(t \times P < t_{max})$ **do**
  if $(t \,\%\, \mathrm{M} \neq 0)$ **then**
    $\mathrm{TFN}(\{\{\mathrm{q},\mathrm{a}\}^{(t-\mathrm{M})}, ..., \{\mathrm{q},\mathrm{a}\}^{t-1}\}; \theta_{\mathrm{TFN}}) \rightarrow \widetilde{\mathrm{v}}^{(t)}, \widetilde{\mathrm{q}}^{(t)}$
    $\mathrm{CAN}(\{\{\mathrm{q},\mathrm{a}\}^{(t-\mathrm{M})}, ..., \{\mathrm{q},\mathrm{a}\}^{t-1}\}, \widetilde{\mathrm{q}}^{(t)}; \theta_{\mathrm{CAN}}) \rightarrow \mathrm{a}^{(t)}$
    $\mathcal{S}(\mathrm{a}^{(t)}) \rightarrow \{\mathrm{q},\mathrm{a}\}^{(t)}$
    $t = t + 1$
  else
    calculate $\{\mathrm{v}^{(t-\mathrm{M})}, ..., \mathrm{v}^{(t)}\}$ via Eq. 5
    optimize $\theta_{\mathrm{TFN}}, \theta_{\mathrm{CAN}}$ by minimizing Eq. 6 and Eq. 7
  end if
end while
return $\theta_{\mathrm{TFN}}, \theta_{\mathrm{CAN}}$

As we discussed above, other agents' newly generated actions will impact the action values and cause action value deviation between perceived and real action values. To overcome this problem, we use the difference between predicted and perceived action value to help anchoring agent action. We optimize the CAN by minimizing the following loss function:

$$\mathcal{L}_{\mathrm{CAN}} = \sum_{t=0}^{\mathrm{M}} \log(\mathrm{a}^{(t)})(\widetilde{\mathrm{v}}^{(t)} - \mathrm{v}^{(t)}). \tag{7}$$

4 Experiment

We conduct experiments in two open-source traffic simulators: CityFlow [1] and SUMO [2]. Both of them are commonly used in previous works [Wei *et al.*, 2018; Wei *et al.*, 2019b; Wei *et al.*, 2019a].

4.1 Datasets

Real-world Datasets

We used three real-world datasets: $D_{Hangzhou}$, D_{Jinan} and $D_{NewYork}$. All three datasets are obtained from CoLight [Wei *et al.*, 2019b]. The New York, Hangzhou and Jinan datasets have 196, 16 and 12 intersections, respectively. We used the same simulator (CityFlow) and experimental settings as CoLight to conducted fair comparison.

Synthetic Datasets

In SUMO simulator, we build a road network with 21 intersections to simulate the urban trunk roads, as shown in Figure 4. The road network is constructed by following Wei *et al.* [2019c]. Intersections in the middle of the road network have four directions, and intersections on the boundary have three

[1] https://cityflow-project.github.io
[2] http://sumo.dlr.de/index.html

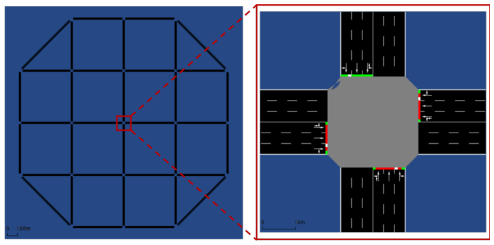


Figure 4: Left: Sketch of the simulated 21 intersections road network [Wei *et al.*, 2019c] we used in synthetic experiments. Right: An intersection example with four directions.

Config	Traffic	Arrival Rate (car/300s)	Trend	Start Time	End Time
1	T1	150	mixed	0	20000
	T2	150	mixed	0	20000
	T3	150	mixed	0	20000
2	T4	120	mixed	0	10800
	T5	180	mixed	10800	18000
	T6	300	mixed	10800	18000
3	T7	150	mixed	0	20000
	T8	210	W→E	0	20000

Table 1: Configurations for simulation traffic. 'W→E' represents that most vehicles are running from west to east on the road network.

directions. Each road is 500 meters long, except roads in the corner is 680 meters. Each road has two directions and three lanes per direction. Road speed limit is set as 16.68 m/s, and the period length is 120 seconds.

As shown in Table 1, we built three different traffic scenarios based on the road network above to mimic the prevailing and challenging traffic conditions in the real world. In Config 1, we synthesized the daytime with heavy flat traffic. In Config 2, we synthesized the traffic switching from light flat-time traffic to heavy peak-time traffic. In Config 3, we simulated the tidal traffic in which part of traffic has a strong trend (West→East) during heavy peak-time traffic. All three kinds of configurations are widespread in metropolitan areas around the world.

The arrival rate defines the traffic density in these configurations. The traffic trend of different traffic groups (T1-T8 in Table 1) is determined by each road's probability of becoming a starting point, via point and destination from a uniform distribution. End time represents the departure time of the last vehicle. All data contain bidirectional and dynamic flows with turning traffic. Moreover, we used the vehicle rerouter algorithm provided by SUMO to synthesize vehicle rerouting cases as in the real world.

Traffic Forecasting Dataset

To evaluate the traffic forecasting performance of TFN, we conduct experiments in the persuasive real-world traffic forecasting dataset METR-LA [Li *et al.*, 2018] comparing with several state-of-the-art methods.

4.2 Compared Methods

We compare MaCAR with several baseline methods, including several state-of-the-art methods.

In synthetic experiments, we compare with four different baseline methods, including Fixed-time Control (FT), Random Adjustment (RA), Actuated Control (SOTL [Cools *et al.*, 2013]), and Q-Learning Traffic Light Optimization within Multiple Intersections Traffic Network (QLTSO) [Chin *et al.*, 2012]. Each baseline represents a type of commonly used method in real-world applications.

Methods	Avg.speed (m/s)	Avg.Queue	Avg.Waiting (s)
FT	4.22	3.08	28.55
RA	5.61	3.19	27.52
SOTL	6.76	2.44	17.64
QLTSO	5.35	3.71	41.76
MaCAR-noTFN	6.62	2.75	17.98
MaCAR	**7.30**	**2.19**	**13.40**

Table 2: Results on Synthetic Dataset Config 1.

Methods	Avg.speed (m/s)	Avg.Queue	Avg.Waiting (s)
FT	7.82	0.81	6.94
RA	8.20	0.81	6.75
SOTL	8.42	0.70	5.32
QLTSO	8.13	0.95	10.19
MaCAR-noTFN	8.29	0.71	5.43
MaCAR	**8.69**	**0.68**	**5.13**

Table 3: Results on Synthetic Dataset Config 2.

We then conduct experiments in three real-world datasets, comparing with several state-of-the-art methods [Wei *et al.*, 2019b; Wei *et al.*, 2018; Chu *et al.*, 2019; Nishi *et al.*, 2018; Arel *et al.*, 2010; Van der Pol and Oliehoek, 2016]. The state-of-the-art method CoLight [Wei *et al.*, 2019b] is a collaborative optimization based method, in which information exchange between agents is carried out by expanding other agents' observable surrounding.

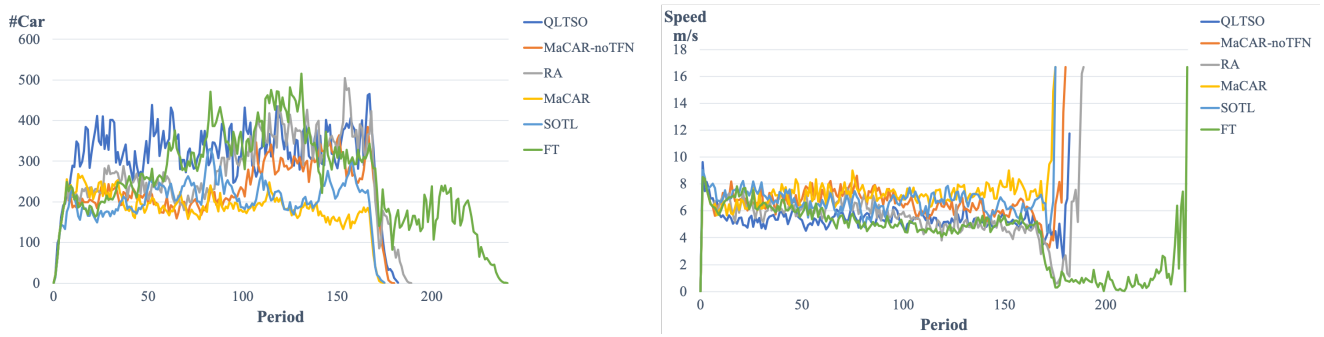
4.3 Experiments on Synthetic Datasets

We first compare our proposal with several baselines under three traffic scenarios, experimental results have shown in Table 2−4. We can notice that MaCAR can effectively improve traffic efficiency under all three different traffic conditions. Meanwhile, our proposal outperforms all baselines significantly in Avg.Waiting in Config 2 and Config 3. It demonstrates that when traffic shifts, our method still can effectively change the phase plans to mitigate traffic congestion.

In Figure 5 (a) and (b), we show that MaCAR can significantly reduce the total queue length and increase the avg. speed. In Figure 5 (b), we can notice the curve of MaCAR is flatter than other methods and has an upward trend even when the traffic starts to increase from 60 periods. Moreover, the upward trend of MaCAR's curve is more significant than MaCAR-noMPTF. These experiments validate our idea of taking advantage of prediction information is feasible while demonstrating the effectiveness of the proposed cross-agent communication mechanism.

4.4 Experiments on Real-world Datasets

We then compare our proposal with several state-of-the-art methods in real-world datasets. We followed the experimental setup used by CoLight to compare our method with previous works fairly. Results have shown in Table 5, MaCAR achieves consistent performance improvements over state-of-the-art methods on all three real-world datasets: the average improvement is 2.78% compare with CoLight. The per-



(a) (b)

Figure 5: (a) Total queue length of the road network under traffic Config 1. The lower the curve, the higher the road patency. (b) The average speed of all vehicles under traffic Config 1. The higher the curve, the higher the road network efficiency.

Methods	Avg.speed (m/s)	Avg.Queue	Avg.Waiting (s)
FT	3.47	4.72	50.88
RA	3.26	4.92	52.46
SOTL	3.18	4.33	47.32
QLTSO	7.25	2.15	15.83
MaCAR-noTFN	7.24	2.66	20.63
MaCAR	**7.83**	**2.07**	**15.58**

Table 4: Results on Synthetic Dataset Config 3.

Methods	$D_{NewYork}$	$D_{Hangzhou}$	D_{Jinan}
CGRL [Van der Pol and Oliehoek, 2016]	2187.12	1582.25	1210.70
NeighborRL [Arel *et al.*, 2010]	2280.92	1053.45	1168.32
GCN [Nishi *et al.*, 2018]	1876.37	768.43	625.66
OneModel [Chu *et al.*, 2019]	1973.11	394.56	728.63
Individual RL [Wei *et al.*, 2018]	-	345.00	325.56
CoLight [Wei *et al.*, 2019b]	1459.28	297.26	291.14
MaCAR	**1425.00 (+2.30%)**	**291.18 (+2.04%)**	**279.49 (+4.00%)**

Table 5: Results on Real-world Datasets w.r.t average travel time.

formance improvements of our method are attributed to the unique design of our proposal.

Our method also outperforms the joint-action modeling method CGRL. To achieve cooperation, CGRL establishes one model to determine the joint actions of two adjacent intersections, and then conducts centralized coordination of the global joint actions. It needs to search a large operating space, and may faces scalability issues. Compare with CGRL, our method has a smaller search space since we still use independent agents.

In experiments on the New York dataset, we show that MaCAR can outperforms state-of-the-arts in a 196 intersections road network. These experiments also demonstrate that MaCAR has good scalability in practice.

4.5 Experiments on Traffic Forecasting Dataset

We then conduct traffic forecasting experiments on METR-LA dataset to demonstrate the traffic forecasting performance of TFN comparing with state-of-the-art traffic forecasting methods. Results have shown in Table 6, TFN can outperforms all the state-of-the-art methods. We can notice that the state-of-the-art methods may perform properly for short-term forecasting, but their long-term predictions are not accurate due to the error accumulation.

Methods	MAE		
	15mins	30mins	60mins
FC-LSTM	3.44	3.77	4.37
STGCN [Yan *et al.*, 2018]	2.87	3.48	4.45
DCRNN [Li *et al.*, 2018]	2.77	3.15	3.60
ST-UNet [Yu *et al.*, 2019]	2.72	3.12	3.55
TFN	**2.68**	**2.99**	**3.28**

Table 6: Traffic forecasting experiment results on METR-LA. MAE metric is compared for different future time steps.

4.6 Ablation Study

We also compared the performance of variants of our proposed method, as shown in Table 2 - 4. By removing TFN, we notice that the performance of MaCAR-noTFN is reduced significantly, but still can outperforms baseline methods in most metrics. These experiments show that our idea of combining traffic forecasting with traffic light control can effectively improve traffic efficiency.

5 Conclusion

In this work, we address the traffic light control problem by proposing a novel Multi-agent Communication and Action Rectification (MaCAR) framework. We conduct extensive experiments on both synthetic and real-world datasets to demonstrate the superior performance of our proposed method over baseline and state-of-the-art methods. Besides, we show in-depth case studies and observations to understand how the proposed method overcomes two shortcomings of previous collaborative optimization methods. Several future directions are worth exploring, global optimization methods that consider local importance, and causal inference based methods.

Acknowledgements

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 61936006, U1909203), in part by the Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

References

[Arel *et al.*, 2010] Itamar Arel, Cong Liu, Tom Urbanik, and Airton G Kohls. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 4(2):128–135, 2010.

[Aslani *et al.*, 2017] Mohammad Aslani, Mohammad Saadi Mesgari, and Marco Wiering. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*, 85:732–752, 2017.

[Battaglia *et al.*, 2018] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[Casas, 2017] Noe Casas. Deep deterministic policy gradient for urban traffic light control. *arXiv preprint arXiv:1703.09035*, 2017.

[Chen *et al.*, 2020] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *AAAI 2020*, 2020.

[Chin *et al.*, 2012] Yit Kwong Chin, Wei Yeang Kow, Wei Leong Khong, Min Keng Tan, and Kenneth Tze Kin Teo. Q-learning traffic signal optimization within multiple intersections traffic network. In *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*, pages 343–348. IEEE, 2012.

[Chu *et al.*, 2019] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[Cools *et al.*, 2013] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. *Self-Organizing Traffic Lights: A Realistic Simulation*, pages 45–55. Springer London, London, 2013.

[Lee *et al.*, 2019] Donghwan Lee, Niao He, Parameswaran Kamalaruban, and Volkan Cevher. Optimization for reinforcement learning: From single agent to cooperative agents. *arXiv preprint arXiv:1912.00498*, 2019.

[Li *et al.*, 2018] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

[Nishi *et al.*, 2018] Tomoki Nishi, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura. Traffic signal control based on reinforcement learning with graph convolutional neural nets. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 877–883. IEEE, 2018.

[Van der Pol and Oliehoek, 2016] Elise Van der Pol and Frans A Oliehoek. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, 2016.

[Wei *et al.*, 2018] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505. ACM, 2018.

[Wei *et al.*, 2019a] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1290–1298, 2019.

[Wei *et al.*, 2019b] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yamin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, 2019.

[Wei *et al.*, 2019c] L. Wei, Z. Yu, Z. Jin, L. Xie, J. Huang, D. Cai, X. He, and X. Hua. Dual graph for traffic forecasting. *IEEE Access*, pages 1–1, 2019.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[Yan *et al.*, 2018] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[Yu *et al.*, 2019] Bing Yu, Haoteng Yin, and Zhanxing Zhu. St-unet: A spatio-temporal u-network for graph-structured time series modeling. *CoRR*, abs/1903.05631, 2019.

[Zheng *et al.*, 2019] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash V. Gayah, Kai Xu, and Zhenhui Li. Diagnosing reinforcement learning for traffic signal control. *CoRR*, abs/1905.04716, 2019.

[Zhou *et al.*, 2018] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.