CoLight: Learning Network-level Cooperation for Traffic Signal Control

Hua Wei[†][*], Nan Xu[‡][*], Huichu Zhang[‡], Guanjie Zheng[†], Xinshi Zang[‡], Chacha Chen[‡],

Weinan Zhang[‡], Yanmin Zhu[‡], Kai Xu[§], Zhenhui Li[†]

[†]Pennsylvania State University, [‡]Shanghai Jiao Tong Univerisity, [§]Shanghai Tianrang Intelligent Technology Co., Ltd
[†]{hzw77, gjz5038, jessieli}@ist.psu.edu, [‡]{xunannancy, chacha1997, wnzhang, yzhu}@sjtu.edu.cn,
[‡]{zhc,xszang}@apex.sjtu.edu.cn, [§]kai.xu@tianrang-inc.com

ABSTRACT

Cooperation among the traffic signals enables vehicles to move through intersections more quickly. Conventional transportation approaches implement cooperation by pre-calculating the offsets between two intersections. Such pre-calculated offsets are not suitable for dynamic traffic environments.

To enable cooperation of traffic signals, in this paper, we propose a model, *CoLight*, which uses graph attentional networks to facilitate communication. Specifically, for a target intersection in a network, *CoLight* can not only incorporate the temporal and spatial influences of neighboring intersections to the target intersection, but also build up index-free modeling of neighboring intersections. To the best of our knowledge, we are the first to use graph attentional networks in the setting of reinforcement learning for traffic signal control and to conduct experiments on the large-scale road network with hundreds of traffic signals. In experiments, we demonstrate that by learning the communication, the proposed model can achieve superior performance against the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; Control methods; • **Applied computing** → *Transportation*.

KEYWORDS

Deep reinforcement learning, traffic signal control, multi-agent system

*Equal contribution.

1 INTRODUCTION

A key question often asked of traffic signal control is "How do traffic signals cooperate between intersections?" Cooperation among intersections is especially important for an urban road network because the actions of signals could affect each other, especially when the intersections are closely connected. Good cooperation among the traffic signals enables the vehicles to move through intersections more quickly.

In the transportation field, a typical way to solve the cooperation problem between intersections is to formulate it as an optimization problem and solve it under certain assumptions (e.g., uniform arrival rate [21, 27] and unlimited lane capacity [24]). Such methods, however, do not perform well because the assumptions do not hold in the real world.

Recently, researchers start to investigate reinforcement learning (RL) techniques for the cooperation of traffic signals. The most common way is to have an RL agent control each intersection and communication is achieved by sharing information among agents [1, 8, 9, 22]. At each time step, the agent observes the traffic condition of the target intersection and its neighboring intersections, and decides an action a to take. After the action is taken, a reward r (often defined as a measure correlated with travel time) is fed back to the agent indicating how good the action a is. Different from conventional approaches, such RL methods avoid making strong assumptions and directly learn good strategies from trials and errors.

Existing RL-based methods still fail to communicate with neighbors in the most efficient way. We propose *CoLight* that improves communication of agents and is scalable to hundreds of intersections. In particular, our work makes the following key contributions:

• *Cooperation through dynamic communication.* Most methods to achieve cooporation are through expanding the observation of the target agent for more comprehensive information. Existing studies [1, 5, 8, 9, 22] tend to select the traffic conditions from adjacent intersections and directly concatenate them with the traffic condition of the target intersection, neglecting the fact that the traffic is changing both temporally and spatially. For example, if intersections A and B are adjacent intersections on a major road, but intersection C is on a side road linked with B. The information from A is more useful to B compared with that from C to B. Besides,

(a) Two target intersections in a real-world road network

(b) Model input alignment
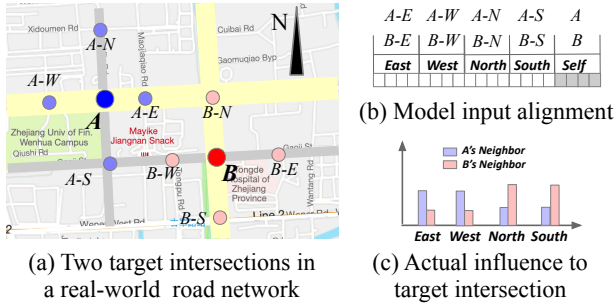
(c) Actual influence to target intersection

Figure 1: Illustration of index-based concatenation. Thick yellow lines are the arterials and grey thin lines are the side streets. With index-based concatenation, _A_ and _B_'s observation will be aligned as model inputs with an fixed order. These two inputs will confuse the model shared by _A_ and _B_.

the influences from intersection _A_ could change temporally. For example, during the morning rush hour, there are a large number of vehicles moving from _A_ to _B_, and the direction of vehicles is reversed at night peak hours. Therefore, the effect of _A_ on _B_ is also changing at different times of the day. In this paper, we propose to use the graph attentional network [26] to learn the dynamics of the influences, which shows superior performance against methods without this mechanism in Section 5.5. To the best of our knowledge, we are the first to use graph attentional network in the setting of traffic signal control.

• _Index-free model learning with parameter sharing._ Agents often share the same model parameters to enable efficient learning [1, 9]. However, if we simply take neighboring information as part of the state for the communication purpose, the fixed indexing of neighbors may cause conflicting learning problems. Take Figure 1 as an example. Figure 1(a) shows two target intersections _A_ and _B_, where their neighbors _A-W_, _A-E_, _B-N_ and _B-S_ are on the major roads, while _A-N_, _A-S_, _B-W_ and _B-E_ are on the side streets. Intuitively, the two intersections on the major roads should relate more to the target intersection than those on the side streets, e.g., the influence from _A-W_ to _A_ should be greater than that from _A-N_ to _A_. When neighbors are indexed in the way as [E, W, N, S], agent _A_ would care more about the first and second neighbors (i.e., East and West) and agent _B_ would pay more attention to the third and fourth neighbors (i.e., North and South). However, when two agents share the model parameters, this will cause conflicts to learn the influences of neighbors to the target intersection. In this paper, similar to the "mean-field" idea [32], we tackle this problem by averaging over the influences of all neighboring intersections with the learned attention weights, instead of using a fixed indexing for neighbors. This weighted average mechanism provides index-free modeling of the influences of neighboring intersections, and along with the parameter sharing strategy, the overall parameters of the learning model can be largely reduced. We will show the parameter analysis later in Section 4.4.

• _Experiment on the large-scale road network._ To the best of our knowledge, none of the existing studies that use RL to cooperate traffic signals have evaluated their methods in large-scale road networks with hundreds of traffic signals. Instead, most of them justify their cooperation strategies on small road networks with

only fewer than 20 intersections [29]. In this paper, we conduct comprehensive experiments on both synthetic and real-world data, including a large-scale road network with 196 intersections derived from Manhattan, New York. The experiments demonstrate that the proposed model benefits from the dynamic communication and the index-free modeling mechanism and significantly outperforms the state-of-the-art methods.

2 RELATED WORK

Conventional coordinated methods [15] and systems [12, 13, 17] in transportation usually coordinate traffic signals through modifying the offset (i.e., the time interval between the beginnings of green lights) between consecutive intersections and require the intersections to have the same cycle length. But this type of methods can only optimize the traffic flow for certain pre-defined directions [10]. Actually, it is not an easy task to coordinate the offsets for traffic signals in the network. For network-level control, Max-pressure [24] is a state-of-the-art signal control method which greedily takes actions that maximizes the throughput of the network, under the assumption that the downstream lanes have unlimited capacity. Other traffic control methods like TUC [7] also use optimization techniques to minimize vehicle travel time and/or the number of stops at multiple intersections under certain assumptions, such as the traffic flow is uniform in a certain time period. However, such assumptions often do not hold in the network setting and therefore prevent these methods from being widely applied.

Recently, reinforcement learning techniques have been proposed to coordinate traffic signals for their capability of online optimization without prior knowledge about the given environment. One way to achieve coordination is through centralized optimization over the joint actions of multiple intersections. [20] directly trains one central agent to decide the actions for all intersections but it cannot learn well due to the curse of dimension in joint action space. [16, 23] propose to jointly model two adjacent intersections and then using centralized coordination over the global joint actions, but they require the maximization over a combinatorially large joint action space and face scalability issues during deployment. Therefore, these centralized methods are hard to apply on the large-scale road network.

To mitigate this issue, independently modelling RL methods [3, 28, 31] are proposed in which they train a bunch of RL agents separately, one for each intersection. To avoid the non-stationary impacts from neighboring agents, communication among agents [2, 19] are proposed in order to achieve coordination using neighboring information: [5, 8, 9, 22, 31] add downstream information into states, [1, 31] add all neighboring states, and [18] adds neighbors' hidden states. However, in these methods, the information from different neighbors is concatenated all together and treated with equal importance, which leads to two major issues: 1). the impacts from neighboring intersections are changing dynamically with the traffic flow and should not be treated evenly. Even when the traffic flow is static, Kinenmatic-wave theory [11] from the transportation area shows that the upstream intersections could have larger influence than downstream intersections; 2). simple concatenation of neighboring information requires all agents to have an extra indexing mechanism, which is usually unrealistic and requires heuristic design. To address the shortcomings of previous

methods, our proposed method leverages the attention mechanism to learn and specify different weights to different neighboring intersections and directly models the overall influence of neighboring intersections on the target intersection.

It is worth noting that most of joint modelling methods for the traffic signal control problem only conduct experiments on simple road networks with at most 20 intersections. To the best of our knowledge, none of the individual modelling methods conduct experiments on more than 70 signals [29]. In this paper, we conduct experiments on the simulator under different scales, including a real-world road network with about 200 intersections.

3 PROBLEM DEFINITION

In this section, we present the problem of traffic signal control as a Markov Game. Each intersection in the system is controlled by an agent. Given each agent observes part of the total system condition, we would like to proactively decide for all the intersections in the system which phases should they change to so as to minimize the average queue length on the lanes around the intersections. Specifically, the problem is characterized by the following major components $\langle \mathcal{S}, O, \mathcal{A}, \mathcal{P}, r, \pi, \gamma \rangle$:

• System state space $\mathcal{S}$ and observation space O. We assume that there are N intersections in the system and each agent can observe part of the system state $s \in \mathcal{S}$ as its observation $o \in O$. In this work, we define o_i^t for agent i at time t, which consists of its current phase (which direction is in green light) represented by a one-hot vector, and the number of vehicles on each lane connected with the intersection.

• Set of actions $\mathcal{A}$. In the traffic signal control problem, at time t, an agent i would choose an action a_i^t from its candidate action set $\mathcal{A}_i$ as a decision for the next Δt period of time. Here, each intersection would choose a phase p as its action a_i^t from its pre-defined phase set, indicating that from time t to $t + \Delta t$, this intersection would be in phase p.

• Transition probability $\mathcal{P}$. Given the system state s^t and corresponding joint actions $\mathbf{a}^t$ of agents at time t, the system arrives at the next state s^{t+1} according to the state transition probability $\mathcal{P}(s^{t+1}|s^t, \mathbf{a}^t) : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \rightarrow \Omega(\mathcal{S})$, where $\Omega(\mathcal{S})$ denotes the space of state distributions.

• Reward r. Each agent i obtains an immediate reward r_i^t from the environment at time t by a reward function $\mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \rightarrow \mathbb{R}$. In this paper, we want to minimize the travel time for all vehicles in the system, which is hard to optimize directly. Therefore, we define the reward for intersection i as $r_i^t = -\Sigma_l u_{i,l}^t$ where $u_{i,l}^t$ is the queue length on the approaching lane l at time t.

• Policy π and discount factor γ. Intuitively, the joint actions have long-term effects on the system, so that we want to minimize the expected queue length of each intersection in each episode. Specifically, at time t, each agent chooses an action following a certain policy $O \times \mathcal{A} \rightarrow \pi$, aiming to maximize its total reward $G_i^t = \Sigma_{t=\tau}^T \gamma^{t-\tau} r_i^t$, where T is total time steps of an episode and $\gamma \in [0, 1]$ differentiates the rewards in terms of temporal proximity.

In this paper, we use the action-value function $Q_i(\theta_n)$ for each agent i at the n-th iteration (parameterized by θ) to approximate the total reward G_i^t with neural networks by minimizing the loss:

$$\mathcal{L}(\theta_n) = \mathbb{E}[(r_i^t + \gamma \max_{a'} Q(o_i^{t\prime}, a_i^{t\prime}; \theta_{n-1}) - Q(o_i^t, a_i^t; \theta_n))^2] \quad (1)$$

where $o_i^{t\prime}$ denotes the next observation for o_i^t. These earlier snapshots of parameters are periodically updated with the most recent network weights and help increase the learning stability by decorrelating predicted and target q-values.

4 METHOD

In this section, we will first introduce the proposed cooperated RL network structure, as Figure 2 illustrates, from bottom to top layer: the first observation embedding layer, the interior neighborhood cooperation layers (shorted as GAT layers) and the final q-value prediction layer. Then we will discuss its time and space complexity compared with other methods of signal control for multiple intersections.

4.1 Observation Embedding

Given the raw data of the local observation, i.e., the number of vehicles on each lane and the phase the signal currently in, we first embed such k-dimensional data into an m-dimensional latent space via a layer of Multi-Layer Perceptron:

$$h_i = Embed(o_i^t) = \sigma(o_i W_e + b_e), \quad (2)$$

where $o_i^t \in \mathbb{R}^k$ is intersection i's observation at time t and k is the feature dimension of o_i^t, $W_e \in \mathbb{R}^{k \times m}$ and $b_e \in \mathbb{R}^m$ are weight matrix and bias vector to learn, σ is ReLU function (same denotation for the following σ). The generated hidden state $h_i \in \mathbb{R}^m$ represents the current traffic condition of the i-th intersection.

4.2 Graph Attention Networks for Cooperation

Communication between agents is necessary for cooperation in multi-agent reinforcement learning (MARL) environment, since the evaluation of the conducted policy for each agent depends not only on the observable surrounding, but also on the policies of other agents [6, 19]. *CoLight* agent learns to communicate on the representations of neighboring intersections by leveraging the attention mechanism, a widely-used technique to boost model accuracy [4, 14, 33, 34]. Then an overall summary of the agent's neighborhood is generated, upon which the agent learns to model the influence of neighborhoods.

4.2.1 Observation Interaction. To learn the importance of information from intersection j (source intersection) in determining the policy for intersection i (target intersection), we first embed the representation of these two intersections from previous layer and calculate $e_{i,j}$, the importance of j in determining the policy for i, with the following operation:

$$e_{ij} = (h_i W_t) \cdot (h_j W_s)^T, \quad (3)$$

where $W_s, W_t \in \mathbb{R}^{m \times n}$ are embedding parameters for the source and target intersection respectively.

Note that e_{ij} is not necessarily equal to e_{ji}. Take the scenario in Figure 2(a) as an example, where the 9-th Avenue is a one-way arterial. On one hand, since the traffic flow goes from *Inter 9-50* to *Inter 9-49*, the traffic condition from *Inter 9-50* is important for *Inter 9-49* to decide for the future actions, thus, $e_{9-49, 9-50}$ should be quite large. On the other hand, as the traffic condition of the downstream *Inter 9-49* is less helpful to *Inter 9-50*, $e_{9-50, 9-49}$ should be relatively small.
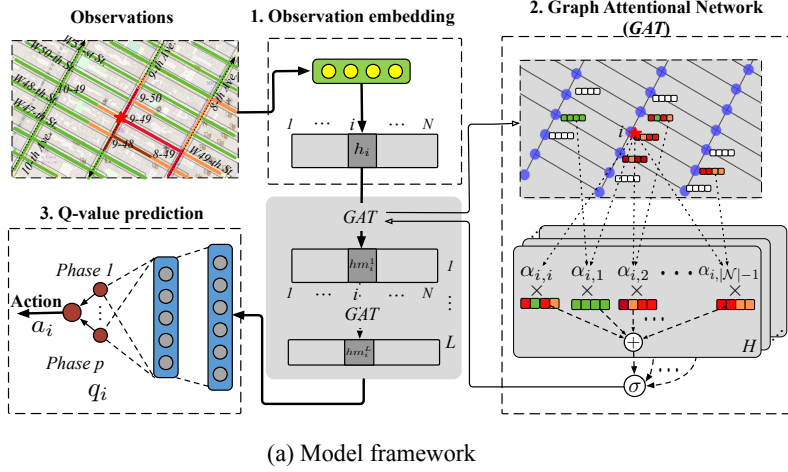
(b) 1st GAT layer (one-hop view)

(a) Model framework

(c) 2nd GAT layer (two-hop view)

Figure 2: Left: Framework of the proposed *CoLight* model. Right: variation of cooperation scope (light blue shadow, from one-hop to two-hop) and attention distribution (colored points, the redder, the more important) of the target intersection.

4.2.2 Attention Distribution within Neighborhood Scope. To retrieve a general attention value between source and target intersections, we further normalize the interaction scores between the target intersection i and its neighborhood intersections:

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij}/\tau)}{\sum_{j \in \mathcal{N}_i} \exp(e_{ij}/\tau)}, \qquad (4)$$

where τ is the temperature factor and $\mathcal{N}_i$ is the set of intersections in the target intersection's *neighborhood scope*. The neighborhood of the target contains the top $|\mathcal{N}_i|$ closest intersections, and the distance can be defined in multiple ways. For example, we can construct the neighborhood scope for target intersection i through:

• *Road distance*: the geo-distance between two intersections' geo-locations.

• *Node distance*: the smallest hop count between two nodes over the network, with each node as an intersection.

Note that intersection i itself is also included in $\mathcal{N}_i$ to help the agent get aware of how much attention should be put on its own traffic condition.

The general attention score α_{ij} is beneficial not only for it applies to all kinds of road network structures (intersections with different numbers of arms), but also for it relaxes the concept of "neighborhood". Without losing generality, the target can even take some other intersections into $\mathcal{N}_i$ although they are not adjacent to them. For instance, one four-way intersection can determine its signal policy based on information from five nearby intersections, four of which are the adjacent neighbors while the other is disconnected but geographically close to the target intersection.

4.2.3 Index-free Neighborhood Cooperation. To model the overall influence of neighborhoods to the target intersection, the representation of several source intersections are combined with their respective importance:

$$hs_i = \sigma\Big(W_q \cdot \sum_{j \in \mathcal{N}_i} \alpha_{ij}(h_j W_c) + b_q\Big), \qquad (5)$$

where $W_c \in \mathbb{R}^{m \times c}$ is weight parameters for source intersection embedding, W_q and b_q are trainable variables. The weighted sum of

neighborhood representation $hs_i \in \mathbb{R}^c$ accumulates the key information from the surrounding environment for performing efficient signal policy. By summing over the neighborhood representation, the model is *index-free* that does not require all agents to align the index of their neighboring intersections.

The graph-level attention allows the agent to adjust their focus according to the dynamic traffic and to sense the environment in a larger scale. In Figure 2(b) and (c), the emphasizes of *Inter 9-49* on four neighbors are quite distinct due to the uni-directional traffic flow, i.e., a higher attention score for *Inter 9-50* (upstream, red marked) than for *Inter 9-48* (downstream, green marked). The agent for *Inter 9-49* acquires the knowledge of adjacent intersections (*Inter 9-48*, *Inter 9-50*, *Inter 10-49* and *Inter 8-49*) directly from the first layer of Graph Attention Networks (GAT). Since the hidden states of adjacent neighbors from the first GAT layer carry their respective neighborhood message, then in the second GAT layer, the cooperation scope of *Inter 9-49* expands significantly (blue shadow in Figure 2(c)) to 8 intersections. Such additional information helps the target *Inter 9-49* learn the traffic trend. As a result, *Inter 9-49* relies more on the upstream intersections and less on the downstream to take actions, and the attention scores on *Inter 9-50* and *Inter 8-49* grow higher while those on *Inter 10-49* and *Inter 9-48* become lower. More GAT layers helps the agent detect environment dynamics more hops away.

4.2.4 Multi-head Attention. The cooperating information hs_i for the i-th intersection concludes one type of relationship with neighboring intersections. To jointly attend to the neighborhood from different representation subspaces at different positions, we extend the previous single-head attention in the neural network to multi-head attention as much recent work did [25, 26]. Specifically, the attention function (procedures including *Observation Interaction*, *Attention Distribution* and *Neighborhood Cooperation*) with different linear projections (multiple sets of trainable parameters $\{W_t, W_s, W_c\}$) is performed in parallel and the different versions of neighborhood condition summarization hs_i are averaged as hm_i:

$$e_{ij}^h = (h_i W_t^h) \cdot (h_j W_s^h)^T \qquad (6)$$

$$\alpha_{ij}^h = \text{softmax}(e_{ij}^h) = \frac{\exp(e_{ij}^h/\tau)}{\sum_{j \in \mathcal{N}_i} \exp(e_{ij}^h/\tau)} \qquad (7)$$

$$hm_i = \sigma\left(W_q \cdot \left(\frac{1}{H}\sum_{h=1}^{h=H}\sum_{j \in \mathcal{N}_i} \alpha_{ij}^h(h_j W_c^h)\right) + b_q\right) \qquad (8)$$

where H is the number of attention heads. Besides averaging operation, concatenating the product of multi-head attention is another feasible way to conclude multiple types of the neighborhood cooperation.

In this work, we investigate the effects of multi-head attention on performance of the proposed model and find that 5 attention heads achieve the best performance.

4.3 Q-value Prediction

As illustrated in Figure 2(a), each hidden layer of model *CoLight* learns the neighborhood representation through methods introduced in Section 4.2. We denote such layerwise cooperation procedure by *GAT*, then the forward propagation of input data in *CoLight* can be formatted as follows:

$$
\begin{aligned}
hi &= & Embed(o_i^t), \\
hm_i^1 &= & GAT^1(h_i), \\
&\cdots& , \\
hm_i^L &= & GAT^L(hm_i^{L-1}), \\
\widetilde{q}(o_i^t) &= & hm_i^L W_p + b_p,
\end{aligned} \qquad (9)
$$

where $W_p \in \mathbb{R}^{c \times p}$ and $b_p \in \mathbb{R}^p$ are parameters to learn, p is the number of phases (action space), L is the number of GAT layers, $\widetilde{q}$ is the predicted q-value.

According to Eq. (1), the loss function for our *CoLight* to optimize the current policy is:

$$L(\theta) = \frac{1}{T}\sum_{t=1}^{t=T}\sum_{i=1}^{i=N}\left(q(o_i^t, a_i^t) - \widetilde{q}(o_i^t; a_i^t, \theta)\right)^2, \qquad (10)$$

where T is the total number of time steps that contribute to the network update, N is the number of intersections in the whole road network, θ represents all the trainable variables in *CoLight*.

As the importance of neighborhood to the target intersection varies spatially and temporally, the proposed attention mechanism is able to help the target agent distinguish among the complex scenarios by considering the traffic condition of any source-target intersection pair.

4.4 Complexity Analysis

Although *CoLight* spares additional parameters to learn the dynamic cooperation from neighborhood, owing to the index-free parameter sharing mechanism, both the time and space it demands are approximately equal to $O(m^2 L)$, which is irrelevant to the number of intersections. Hence *CoLight* is scalable even if the road network contains hundreds of or even thousands of intersections.

4.4.1 Space Complexity. If there are L hidden layers and each layer has m neurons, then the size of the weight matrices and bias vectors in each component of *CoLight* is: 1) *Observation Embedding* layer: $km + m$; 2) interior *Graph Attentional* layers: $(3m^2 + (m^2 + m))L = m(4m + 1)L$; 3) *Q-value Prediction* layer: $mp + p$. Hence the total

number of learnable parameters to store is $O\big(m(4mL + L + k + 1 + p) + p\big)$. Normally, the size of the hidden layer (m) is far greater than the number of layers (L), the phase space (p) and comparable to the input dimension (k). Therefore, the space complexity of *CoLight* is approximately equal to $O(m^2 L)$.

If we leverage N separate RL models (without parameter sharing) to control signals in N intersections, then the space complexity is $O\big(((km + m) + (m^2 + m)L + (mp + p)) \cdot N\big) \approx O(m^2 L \cdot N)$, which is unfeasible when N is extremely large for city-level traffic signal control. To scale up, the simplest solution is to allow all the intersections to share parameters and maintain one model, in this case, the space complexity is $O(m^2 L)$, which is identical to that of *CoLight*.

4.4.2 Time Complexity. We assume that: 1) all the agents leverage *CoLight* to predict q-values for the corresponding intersections concurrently; 2) the multiple heads of attention are independently computed so that they are as fast as the single-head attention; 3) the embeddings for either source or target intersection condition via W_s, W_c and W_t are separate processes that can also be executed at the same time, 4) for one target intersection, the interaction with all the neighbors is computed simultaneously, then the time complexity (only multiplication operations considered since the addition procedures are relatively insignificant) in each component of *CoLight* is: 1) *Observation Embedding* layer: km; 2) interior *Graph Attentional* layers: $(m^2 + m^2)L$; 3) *Q-value Prediction* layer: mp. Hence the time complexity is $O\big(m(k + 2mL + p)\big)$, and similarly, it is approximately equal to $O(m^2 L)$.

Either the individual RL models or the shared single RL model for signal control in multiple intersections requires $O\big(m(k + mL + p)\big) \approx O(m^2 L)$ computation, approaching that of *CoLight*.

5 EXPERIMENTS

We perform experiments on two synthetic datasets and three real-world datasets to evaluate our proposed method, especially the graph-level attention for neighborhood cooperation.

5.1 Settings

We conduct experiments on CityFlow[1], an open-source traffic simulator that supports large-scale traffic signal control [35]. After the traffic data being fed into the simulator, a vehicle moves towards its destination according to the setting of the environment. The simulator provides the state to the signal control method and executes the traffic signal actions from the control method. Following the tradition, each green signal is followed by a three-second yellow signal and two-second all red time.[2]

In a traffic dataset, each vehicle is described as (o, t, d), where o is the origin location, t is time, and d is the destination location. Locations o and d are both locations on the road network.

5.2 Datasets

5.2.1 Synthetic Data. In the experiment, we use two kinds of synthetic data, i.e., uni- and bi-directional traffic, on the following

---

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

[2]*CoLight*'s codes, parameter settings, public datasets can be found at: https://github.com/wingsweihua/colight. More datasets can be found at: http://traffic-signal-control.github.io

(a) Upper East Side, Manhattan,
New York, USA

(b) Gudang Sub-district,
Hangzhou, China
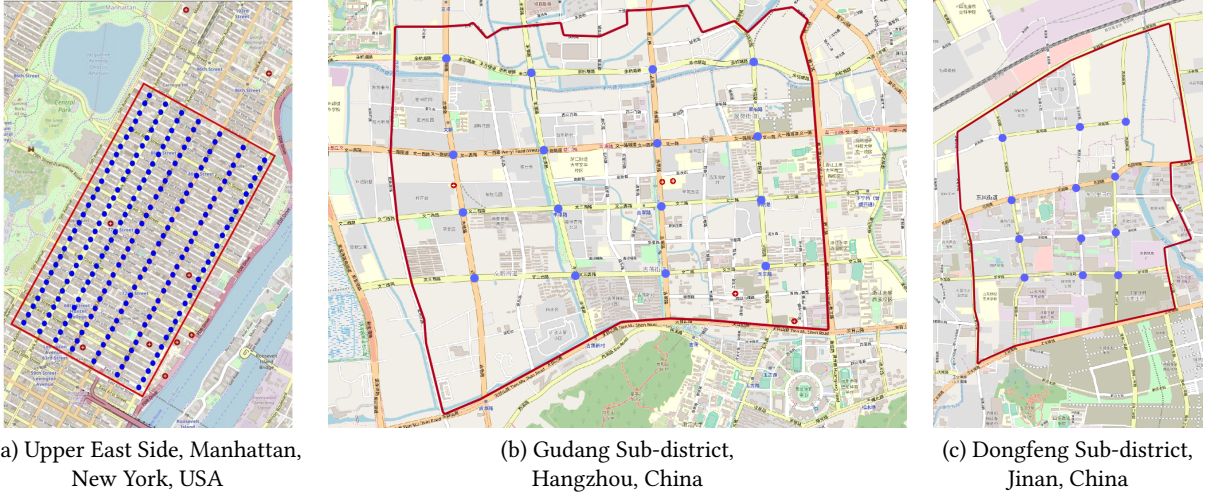
(c) Dongfeng Sub-district,
Jinan, China

Figure 3: Road networks for real-world datasets. Red polygons are the areas we select to model, blue dots are the traffic signals we control. Left: 196 intersections with uni-directional traffic, middle: 16 intersections with uni- & bi-directional traffic, right: 12 intersections with bi-directional traffic.

different road networks:

- $Arterial_{1\times3}$: A 1×3 arterial to show the spatial attention distribution learned by *CoLight*.
- $Grid_{3\times3}$: A 3×3 grid network to show convergence speed of different RL methods and the temporal attention distribution.
- $Grid_{6\times6}$: A 6×6 grid network to evaluate effectiveness and efficiency of different methods.

Each intersection in the synthetic road network has four directions (West→East, East→West, South→North, North→South), and 3 lanes (300 meters in length and 3 meters in width) for each direction. In bi-directional traffic, vehicles come uniformly with 300 vehicles/lane/hour in West↔East direction and 90 vehicles/lane/hour in South↔North direction. Only West→East and North→South directional flows travel in uni-directional traffic.

5.2.2 Real-world Data. We also use the real-world traffic data from three cities: New York, Hangzhou and Jinan. Their road networks are imported from OpenStreetMap[1], as shown in Figure 3. And their traffic flows are processed from multiple sources, with data statistics listed in Table 1. The detailed descriptions on how we preprocess these datasets are as follows:

Table 1: Data statistics of real-world traffic dataset

Dataset	# intersections	Arrival rate (vehicles/300s)			
		Mean	Std	Max	Min
$D_{NewYork}$	196	240.79	10.08	274	216
$D_{Hangzhou}$	16	526.63	86.70	676	256
D_{Jinan}	12	250.70	38.21	335	208

- $D_{NewYork}$: There are 196 intersections in Upper East Side of Manhattan with open source taxi trip data[2]. Since the taxi data only contains the origin and destination geo-locations of each trip, we first map these geo-locations to the intersections and find the

shortest path between them. Then we take the trips that fall within the selected areas.

- $D_{Hangzhou}$: There are 16 intersections in Gudang Sub-district with traffic data generated from the roadside surveillance cameras. Each record in the dataset consists of time, camera ID and the information about vehicles. By analyzing these records with camera locations, the trajectories of vehicles are recorded when they pass through road intersections. We use the number of vehicles passing through these intersections as traffic volume for experiments.
- D_{Jinan}: Similar to $D_{Hangzhou}$, this traffic data is collected by roadside cameras near 12 intersections in Dongfeng Sub-district, Jinan, China.

5.3 Compared Methods

We compare our model with the following two categories of methods: conventional transportation methods and RL methods. Note that all the RL models are learned without any pre-trained parameters for fair comparison.

Transportation Methods:
- *Fixedtime* [15]: Fixed-time with random offsets. This method uses a pre-determined plan for cycle length and phase time, which is widely used when the traffic flow is steady.
- *MaxPressure* [24]: A state-of-the-art network-level traffic signal control method in the transportation field, which greedily chooses the phase that maximizes the pressure (a pre-defined metric about upstream and downstream queue length).

RL Methods:
- *CGRL* [23]: A RL-based method for multi-intersection signal control with joint-action modelling [23]. Specifically, the cooperation is achieved by designing a coordination graph and it learns to optimize the joint action between two intersections.
- *Individual RL* [30]: An individual deep RL approach which does not consider neighbor information. Each intersection is controlled by one agent, and the agents do not share parameters, but update their own networks independently.
- *OneModel* [5]: This method uses the same state and reward as

Table 2: Performance on synthetic data and real-world data w.r.t average travel time. *CoLight* is the best.

Model	$Grid_{6\times6}$-Uni	$Grid_{6\times6}$-Bi	$D_{NewYork}$	$D_{Hangzhou}$	D_{Jinan}
Fixedtime [15]	209.68	209.68	1950.27	728.79	869.85
MaxPressure [24]	186.07	194.96	1633.41	422.15	361.33
CGRL [23]	1532.75	2884.23	2187.12	1582.26	1210.70
Individual RL [30]	314.82	261.60	-*	345.00	325.56
OneModel [5]	181.81	242.63	1973.11	394.56	728.63
Neighbor RL [1]	240.68	248.11	2280.92	1053.45	1168.32
GCN [18]	205.40	272.14	1876.37	768.43	625.66
CoLight-node	178.42	176.71	1493.37	331.50	340.70
CoLight	**173.79**	**170.11**	**1459.28**	**297.26**	**291.14**

*No result as *Individual RL* can not scale up to 196 intersections in New York's road network.

Individual RL in its agent design, which only considers the traffic condition on the roads connecting with the controlled intersection. Instead of maintaining their own parameters, all the agents share the same policy network.

• *Neighbor RL* [1]: Based on *OneModel*, agents concatenate their neighboring intersections' traffic condition with their own and all the agents share the same parameters. Hence its feature space for observation is larger than *OneModel*.

• *GCN* [18]: A RL based traffic signal control method that uses a graph convolutional neural network to automatically extract the traffic features of adjacent intersections. This method treats each neighboring traffic condition without difference.

Variants of Our Proposed Method:

• *CoLight*: The neighborhood scope of an intersection is constructed through geo-distance.

• *CoLight-node*: The neighborhood scope is constructed through node distance, i.e., the smallest hop count between two intersections in the road network.

5.4 Evaluation Metric

Following existing studies [28, 30], we use the average **travel time** to evaluate the performance of different models for traffic signal control. It calculates the average travel time of all the vehicles spend between entering and leaving the area (in seconds), which is the most frequently used measure of performance to control traffic signal in the transportation field [21, 29].

5.5 Performance Comparison

In this section, we investigate on the performance of *CoLight* w.r.t. the travel time and compare it with state-of-the-art transportation and RL methods.

5.5.1 Overall Analysis. Table 2 lists the performance of two types of the proposed *CoLight*, classic transportation models as well as state-of-the-art learning methods in both synthetic and real-world datasets. We have the following observations:

1. *CoLight* achieves consistent performance improvements over state-of-the-art transportation (*MaxPressure*) and RL (*Individual RL*) methods across diverse road networks and traffic patterns: the average improvement is 6.98% for synthetic data and 11.69% for real-world data. The performance improvements are attributed to

the benefits from dynamic communication along with the index-free modeling. The advantage of our model is especially evident when controlling signals in real-world cities, where road structures are more irregular and traffic flows are more dynamic. Specifically, *Individual RL* can hardly achieve satisfactory results because it independently optimizes the single intersection's policy; *Neighbor RL* and *GCN* do not work well for either $D_{NewYork}$ or $D_{Hangzhou}$, as the agent treats the information from the upstream and downstream intersections with static importance according to the prior geographic knowledge rather than real-time traffic flows.

2. The performance gap between the proposed *CoLight* and the conventional transportation method *MaxPressure* becomes larger as the evaluated data change from synthetic regular traffic (average gap 8.08%) to real-world dynamic traffic (average gap 19.89%). Such growing performance divergence conforms to the deficiency inherent in *MaxPressure*, that it is incapable of learning from the feedback of the environment.

3. Our method outperforms the joint-action modelling method *CGRL*. In order to achieve cooperation, *CGRL* first builds up one model to decide the joint actions of two adjacent intersections and then uses centralized coordination over the global joint actions. It requires the centralized maximization over a combinatorially large joint action space and faces scalability issues. On the constrast, our method achieves cooperation through communication between decentralized agents, which has a smaller action space and shows superior performances.

5.5.2 Convergence Comparison. In Figure 4, we compare *CoLight*'s performance (average travel time for vehicles evaluated at each episode) to the corresponding learning curves for the other five RL methods. Evaluated in all the listed datasets, the performance of *CoLight* is better than any of the baselines by a large margin, both in *jumpstart performance* (initial performance after the first episode), *time to threshold* (learning time to achieve a pre-specified performance level), as well as in *asymptotic performance* (final learned performance). Learning the attention on neighborhood does not slow down model convergence, but accelerates the speed of approaching the optimal policy instead.

From Figure 4(a), we discover that model *Individual RL* starts with extremely huge travel time and approaches to the optimal performance after a long training time. Such disparity of convergence speed shown in Figure 4 agrees with our previous space complexity analysis (in Section 4.4.1), that agents with shared models (*CGRL*,
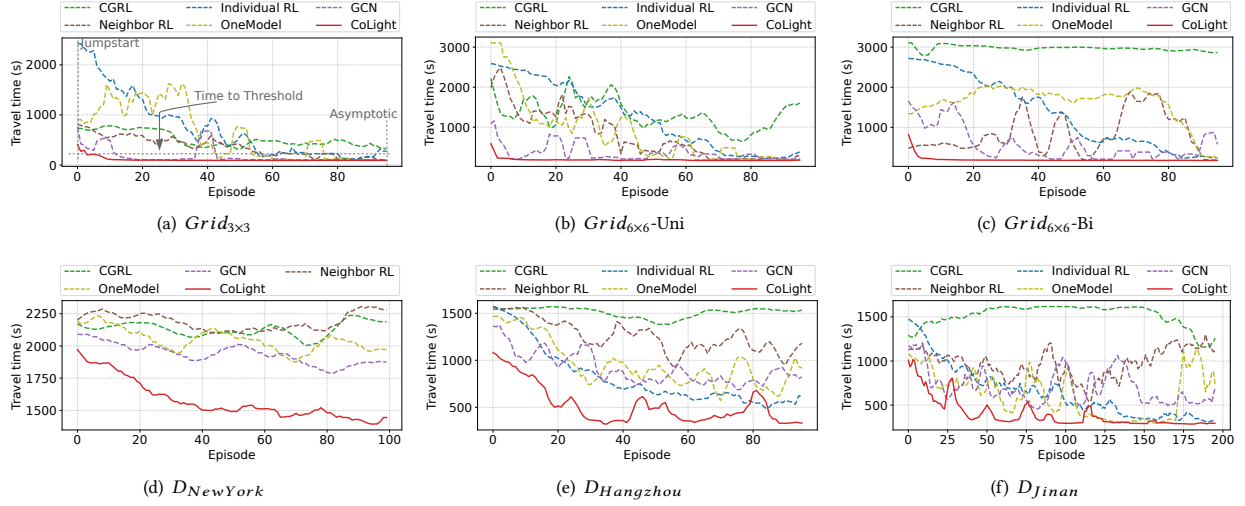
Figure 4: Convergence speed of *CoLight* (red continuous curves) and other 5 RL baselines (dashed curves) during training. *CoLight* starts with the best performance (Jumpstart), reaches to the pre-defined performance the fastest (Time to Threshold), and ends with the optimal policy (Aysmptotic). Curves are smoothed with a moving average of 5 points.

Neighbor RL, *OneModel*, *GCN* and *CoLight*) need to learn $O(m^2L)$ parameters while individual agents (*Individual RL*) have to update $O(m^2L \cdot N)$ parameters.

5.6 Scalability Comparison

In this section, we investigate on whether *CoLight* is more scalable than other RL-based methods in the following aspects:

5.6.1 Effectiveness. As is shown in Table 2 and the convergence curve in Figure 4, *CoLight* performs consistently better than other RL methods on networks with different scales, ranging from 9-intersection grid network to 196-intersection real-world network.

5.6.2 Training time. We compare *CoLight*'s training time (total clock time for 100 episodes) to the corresponding running time for the other five RL methods on road networks with different scales. All the methods are evaluated individually on the server for fair comparison. As is shown in Figure 5, the training time for *CoLight* is comparable to that of *OneModel* and *GCN*, which is far more efficient than that of *CGRL*, *Individual RL* and *Neighbor RL*. This is consistent with the time complexity analysis (in Section 4.4.2), as most of the parallel computation assumptions are satisfied in our experiments.

Note that the average travel time for *Individual RL* (in Table 2) is missing and the bar of training time (in Figure 5) is estimated on $D_{NewYork}$ setting. This is because *Individual RL* is non-scalable because all the separate 196 agents cannot be trained and updated simultaneously due to processor and memory limitation.

5.7 Study of *CoLight*

In this section, we investigate on how different components (i.e., neighborhood definition, number of neighbors, and number of attention heads) affect *CoLight*.

5.7.1 Impact of Neighborhood Definition. As mentioned in Section 4.2.2, the neighborhood scope of an intersection can be defined
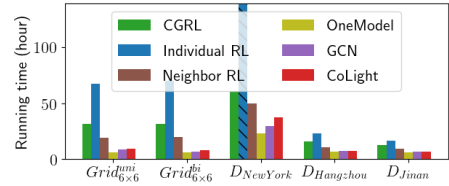

Figure 5: The training time of different models for 100 episodes. *CoLight* is efficient across all the datasets. The bar for *Individual RL* on $D_{NewYork}$ is shadowed as its running time is far beyond the acceptable time.

in different ways. And the results in Table 2 show that *CoLight* (using geo-distance) achieves similar performance with *CoLight-node* under synthetic data, but largely outperforms *CoLight-node* under real-world traffic. The reason could be that under synthetic data, since the lane lengths of all intersections are the same, the top closest neighboring intersections set according to geo-distance is identical to that based on node distance. In the following parts of our experiments, we only compare *CoLight* with other methods.
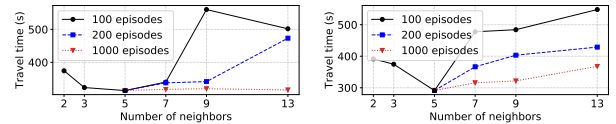

Figure 6: Performance of *CoLight* with respect to different numbers of neighbors ($|\mathcal{N}_i|$) on dataset $D_{Hangzhou}$ (left) and D_{Jinan} (right). More neighbors ($|\mathcal{N}_i| \leq 5$) for cooperation brings better performance, but too many neighbors ($|\mathcal{N}_i| > 5$) requires more time (200 episodes or more) to learn.

5.7.2 Impact of Neighbor Number. In Figure 6, we show how the number of neighbors $|\mathcal{N}_i|$ influences the performance and also

shed lights on how to set it. As is shown in Figure 6, when the number of neighbors grows from 2 to 5, the performance of *CoLight* achieves the optimal. Further adding nearby intersections into the neighborhood scope $\mathcal{N}_i$, however, leads to the opposite trend. This is because including more neighbors in the neighborhood results in learning more relations. To determine signal control policy for each intersection, computing the attention scores only on four nearby intersections and itself seems adequate for cooperation with both time and performance guarantee.

5.7.3 Impact of Attention Head Number. To evaluate the effectiveness of multi-head attention, we test different numbers of attention heads and find that moderate numbers of heads are beneficial to better control intersection signals. As shown in Table 3, drivers spend less time as the number of attention heads grows. However, the benefits of more types of attention disappear as H exceeds 5. Similar conclusions can be made on other datasets with details unshown due to space limitation.

Table 3: Performance of *CoLight* with respect to different numbers of attention heads (H) on dataset $Grid_{6\times6}$. More types of attention ($H \leq 5$) enhance model efficiency, while too many ($H > 5$) could distract the learning and deteriorate the overall performance.

#Heads	1	3	5	7	9
Travel Time (s)	176.32	172.47	170.11	174.54	174.51

6 ATTENTION STUDY

To analyze how well the neighborhood cooperation is implemented via the attention mechanism, we will study both the *spatial* and *temporal* distribution of attention learned by *CoLight* on both synthetic and real-world data.

6.1 Spatial Distribution

The spatial distribution of attention scores indicates the importance of different neighbors to the target agent. In this section, we analyze the learned average attention that *CoLight* learns from the real-world data in each episode. We have the following observations:

•*Upstream vs. Downstream.* Figure 7(a) shows an intersection (green dot) in New York, whose neighborhood includes four nearby intersections along the arterial. Traffic along the arterial is uni-directional (blue arrow). From the attention distribution learned by *CoLight*, we can see that while the majority of attention is allocated to itself, the upstream intersections (orange and blue lines) have larger scores than downstream intersections (red and purple lines).
•*Arterial vs. Side Street.* Figure 7(b) shows an intersection (green dot) in Hangzhou, whose neighborhood includes two intersections along the arterial and two intersections on the side street. In the left part of Figure 7(b), the arterial traffic is heavy and uni-directional (thick horizontal arrow), and the side-street traffic is light and bi-directional (thin vertical arrow). From the right part of Figure 7(b), we can see that the arterial intersections (orange and blue lines) have larger scores than side-street intersections (red and purple lines).

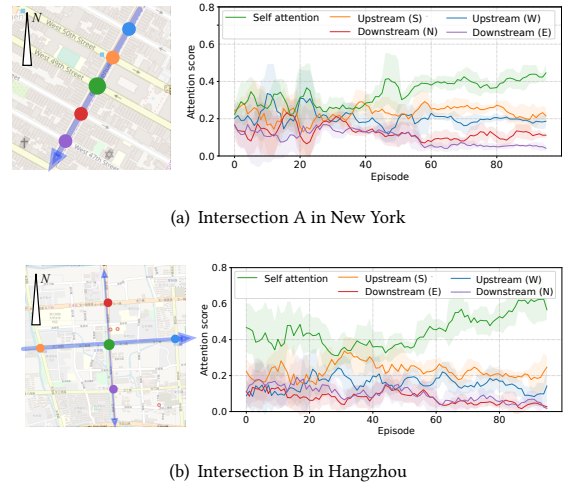(a) Intersection A in New York

(b) Intersection B in Hangzhou

Figure 7: Spatial difference of attention distribution learned by *CoLight* during training process in real-world traffic. Different colored lines in the right figures correspond with the colored dots in the left figures. Up: For intersection A in $D_{NewYork}$, major concentration is allocated on upstream intersections and A itself. Down: For intersection B in $D_{Hangzhou}$, major concentration is allocated on arterial intersections and B itself.

6.2 Temporal Distribution

The temporal distribution of neighbors' attention scores throughout a certain time period indicates the temporal change of neighbors' influence. In this section, we analyze the learned attention of a converged *CoLight* model in a 3×3 network under a 3600-second traffic. As is shown in the upper part of Figure 8(b), there is small basic traffic on every direction; at the same time, similar to the traffic change between morning peak hours and evening peak hours, the traffic of *Inter #4* experiences great changes, where the traffic from *Inter #3* to *Inter #4* drops (blue arrows in Figure 8(a)) and the traffic from *Inter #1* to *Inter #4* increases (green arrows in Figure 8(a)).
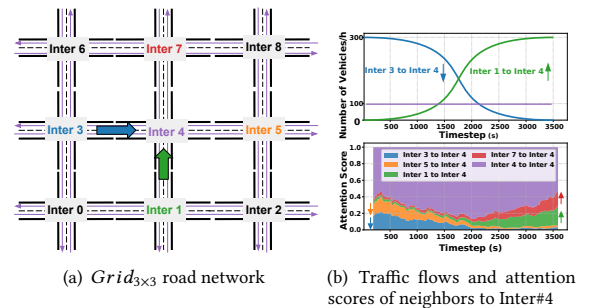

(a) $Grid_{3\times3}$ road network

(b) Traffic flows and attention scores of neighbors to Inter#4

Figure 8: Temporal distribution of attention score learned by *CoLight* corresponds with temporally changing traffic.

The attention scores of the neighboring four intersections of *Inter #4* are shown in the lower part of Figure 8(b). Firstly, we can see that the score from *Inter #4* to *Inter #4* always occupies the

largest purple area, indicating that the traffic condition of *Inter #4* is of a great importance. Secondly, the attention scores of *Inter #1* and *#7* increase with more traffic on South↔North direction. Similarly, the attention scores of *Inter #3* and *#5* decrease as less traffic on West↔East direction. The temporal change of attention scores of *Inter #4*'s four neighbors matches exactly with the change of traffic flow from the four directions, which demonstrates that the proposed *CoLight* model is capable to capture the key neighborhood information over time, i.e., the temporal attention.

7 CONCLUSION

In this paper, we propose a well-designed reinforcement learning approach to solve the network-level traffic signal control problem. Specifically, our method learns the dynamic communication between agents and constructs an index-free model by leveraging the graph attention network. We conduct extensive experiments using synthetic and real-world data and demonstrate the superior performance of our proposed method over state-of-the-art methods. In addition, we show in-depth case studies and observations to understand how the attention mechanism helps cooperation.

We would like to point out several important future directions to make the method more applicable to the real world. First, the neighborhood scope can be determined in a more flexible way. The traffic flow information between intersections can be utilized to determine the neighborhood. Second, the raw data for observation only include the phase and the number of vehicles on each lane. More exterior data like the road and weather condition might help to boost model performance.

ACKNOWLEDGMENTS

REFERENCES

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

[2] Christopher M Bishop et al. 2006. Pattern recognition and machine learning (information science and statistics). (2006).

[3] Eduardo Camponogara and Werner Kraus. 2003. Distributed learning agents in urban traffic control. In *Portuguese Conference on Artificial Intelligence*. Springer, 324–335.

[4] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[5] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[6] Denise de Oliveira and Ana LC Bazzan. 2009. Multiagent learning on traffic lights control: effects of using shared information. In *Multi-agent systems for traffic and transportation engineering*. IGI Global, 307–321.

[7] Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. 2002. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice* 10, 2 (2002), 183–195.

[8] Kurt Dresner and Peter Stone. 2006. Multiagent traffic management: Opportunities for multiagent learning. In *Learning and Adaption in Multi-Agent Systems*. Springer, 129–138.

[9] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1140–1150.

[10] Nathan H Gartner, Susan F Assman, Fernando Lasaga, and Dennis L Hou. 1991. A multi-band approach to arterial traffic signal optimization. *Transportation Research Part B: Methodological* 25, 1 (1991), 55–74.

[11] Wallce D Hayes. 1970. Kinematic wave theory. *Proc. R. Soc. Lond. A* 320, 1541 (1970), 209–226.

[12] PB Hunt, DI Robertson, RD Bretherton, and M Cr Royle. 1982. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control* 23, 4 (1982).

[13] PB Hunt, DI Robertson, RD Bretherton, and RI Winton. 1981. *SCOOT-a traffic responsive method of coordinating signals*. Technical Report.

[14] Jiechuan Jiang, Chen Dun, and Zongqing Lu. 2018. Graph Convolutional Reinforcement Learning for Multi-Agent Cooperation. *arXiv preprint arXiv:1810.09202* (2018).

[15] Peter Koonce et al. 2008. *Traffic signal timing manual*. Technical Report. United States. Federal Highway Administration.

[16] Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. 2008. Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 656–671.

[17] PR Lowrie. 1992. SCATS–a traffic responsive method of controlling urban traffic. Roads and traffic authority. *NSW, Australia* (1992).

[18] Tomoki Nishi, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura. 2018. Traffic Signal Control Based on Reinforcement Learning with Graph Convolutional Neural Nets. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 877–883.

[19] Ann NowÃľ, Peter Vrancx, and Yann MichaÃŕl De Hauwere. 2012. *Game Theory and Multi-agent Reinforcement Learning*.

[20] LA Prashanth and Shalabh Bhatnagar. 2011. Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems* 12, 2 (2011), 412–421.

[21] Roger P Roess, Elena S Prassas, and William R McShane. 2004. *Traffic engineering*. Pearson/Prentice Hall.

[22] Bruno Castro Da Silva, Eduardo W. Basso, Filipo Studzinski Perotto, Ana L. C. Bazzan, and Paulo Martins Engel. 2006. Improving reinforcement learning with context detection. In *International Joint Conference on Autonomous Agents Multiagent Systems*.

[23] van der Pol et al. 2016. Coordinated Deep Reinforcement Learners for Traffic Light Control. NIPS.

[24] Pravin Varaiya. 2013. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, 27–66.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[26] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* 1, 2 (2017).

[27] Fo Vo Webster. 1958. *Traffic signal settings*. Technical Report.

[28] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. 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)*. ACM, 1290–1298.

[29] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A Survey on Traffic Signal Control Methods. *CoRR* abs/1904.08117 (2019). arXiv:1904.08117 http://arxiv.org/abs/1904.08117

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

[31] MA Wiering. 2000. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*. 1151–1158.

[32] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean Field Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:1802.05438* (2018).

[33] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling Spatial-Temporal Dynamics for Traffic Prediction. *arXiv preprint arXiv:1803.01254* (2018).

[34] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.

[35] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. (2019).