

# State Encoding for Efficient Traffic Signal Control in High Volume

Weibin Zhang<sup>a</sup>, Chao Wan<sup>a</sup> and Shoufeng Lu<sup>b</sup>

<sup>a</sup>School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China; <sup>b</sup>School of Transportation Engineering, Nanjing Tech University, Nanjing, China

## ARTICLE HISTORY

Compiled September 8, 2024

## ABSTRACT

In this study, the potential of reinforcement learning (RL) for tackling traffic signal control (TSC) problems is explored, focusing on the challenges posed by large state spaces and limited simulation resources. We transform sparse traffic state evaluation metrics into optimizable dense metrics. By comparing the distribution of states accessed by different control methods, a basis for solving the vast state space is then identified. Building on this, we developed an encoding function to compress regions that are less relevant to the TSC model. Additionally, potential relationships between different states are learned to compensate for information loss due to state space compression, which also play a role in facilitating cooperation among different intersections. Finally, the spacetime dependencies among different intersections are revisited to improve the generalizability of this approach. Experiments with real-world datasets demonstrate that the encoding function can directly reduce unnecessary exploration and achieve superior performance compared to unencoded models.

## KEYWORDS

Traffic signal control; reinforcement learning; state space; state compression

## 1. Introduction

The complexity of traffic systems increases with urban development Feng et al. (2024), leading to a series of problems such as congestion, environmental pollution, and energy waste. These issues significantly impact the quality of life for urban residents and the sustainable development of cities. Hence, an efficient signal control system is crucial for improving urban transportation efficiency.

Traditional signal control systems still occupy a proportion in current TSC systems due to their stability and ease of implementation Wei et al. (2019c). These systems rely on fixed timing plans or simple responsive strategies that poorly adapt to dynamic traffic conditions, which also fail to facilitate the collaborative optimization of traffic across intersections Wei et al. (2019b). In light of these challenges, there is a growing interest in exploring new methodologies that can dynamically adapt to complex traffic systems and offer more effective solutions.

RL has emerged as a novel paradigm for addressing TSC problems, which can

maximize a specified reward function through interactions between the agent and the environment Wei and Zheng (2021); Mnih et al. (2015). A key advantage of RL for signal control is its ability to achieve global optimization through cooperation among agents. Furthermore, RL possesses a certain level of generalization capacity, allowing for rapid deployment on road networks of various structures for optimization purposes Wang et al. (2023). However, the inherent limitations of RL and the complexity of the traffic environment leave several open questions for further exploration.

The first issue is the immense state space caused by the complex traffic environment. RL maximizes the reward function by searching for optimal trajectory within the state space. The vast state space may mislead the direction of searching for the optimal solution. Efficient-CoLight has proved that optimizing traffic state representations with RL-based models could yield significant improvements, especially in large-scale TSC problems Zhang et al. (2021). Furthermore, exponential lower bounds will be accessible for valued-based, model-based, and policy-based algorithms with given good representations Du et al. (2019). Such representations lead to a smaller state space to be explored, but this may result in critical information loss. A balance between the state space and information retention is essential. Hence, this study proposes a biased state encoding function designed to retain key information while minimizing the state space that needs to be explored.

The second issue is the difficulty in characterizing the spacetime dependencies between different intersections due to the interconnected nature of traffic flow. In a road network, the flow of vehicles represents the interactions between intersections, influenced by their temporal and spatial relationships. As mentioned above, traditional methods struggle to capture such relationships. In RL, these relationships can be modeled through the sharing of partial state information Wei et al. (2019b); Oroojlooy et al. (2020); Chen et al. (2022). Furthermore, by sharing partial information, local agents can understand adjacent agents, thereby achieving the purpose of co-operation. Nevertheless, simple information sharing is also inadequate to completely capture the state of traffic flow Cao et al. (2020). Therefore, this paper proposes a spacetime dependency capture model based on the fusion of relevant states to provide a comprehensive understanding of traffic dynamics.

In summary, the contributions of this study are as follows:

- By utilizing a biased encoding function, the state space is selectively compressed. The model aims to preserve information in critical regions while compressing the state space in lesser relevant areas.
- A spacetime dependency capture model based on fusing relevant states between intersections is proposed. By extracting latent information from related states, the model more accurately characterizes traffic dynamics and provides a basis for cooperation between different intersections.

## 2. RELATED WORKS

In this section, we review the key developments in addressing TSC problems, which will be divided into two main categories: the traditional approach, and the RL-based approach. The latter is further categorized into value-based algorithms and Actor-Critic (AC) algorithms Gao et al. (2017); Farebrother, Machado, and Bowling (2018); Schulman et al. (2017); Chu et al. (2019). There are also some hierarchical models for solving signal control problems, but they introduce more complex traffic objects and

are therefore not involved Liu et al. (2021); Huang et al. (2023). Furthermore, we also review works related to state space, which contributes to a better resolution of TSC problems.

The Webster algorithm calculates a set of fixed timing schemes based on historical data and expert experience, which is effective for stable traffic patterns but falls short in adapting to dynamic traffic conditions, still plays a role in offline signal timing calculation scenarios Webster (1958). With the growing complexity and unpredictability of traffic situations, its performance becomes insufficient. Therefore, some algorithms capable of real-time feedback on traffic flow conditions, such as SOTL and SCATS Koonce et al. (2008); Lowrie (1990); Cools, Gershenson, and D’Hooghe (2006). In conjunction with sensing devices, these algorithms achieve local real-time optimization of timing schemes. However, with the expanding scope of signal timing schemes, the need for coordinated optimization across intersections has become increasingly apparent. Given these challenges, the application of RL techniques to TSC problems offers a promising approach for adaptive and cooperative signal optimization.

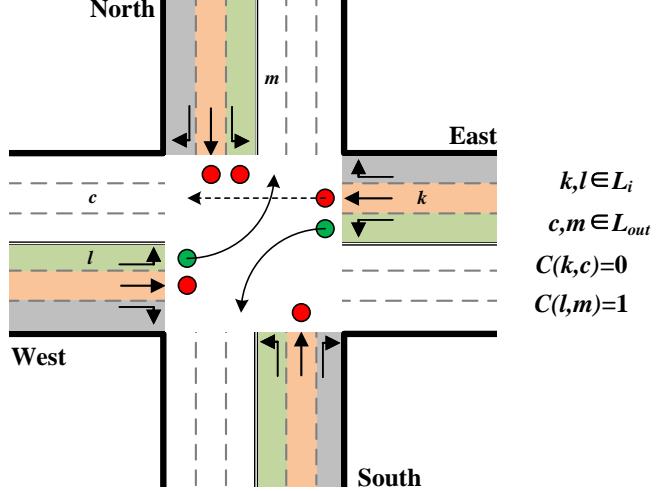
In the RL paradigm, the model continuously optimizes through a scalar generated by a predefined reward form. This optimization has various forms, and in TSC problems, value-based algorithms and AC algorithms are widely used. Value-based algorithms are advantageous due to their efficiency in data utilization, which allows effective use of historical data without the need for memory clearing after an update. CoLight, FRAP, OAM, and others are successful examples of value-based reinforcement learning methods Liang et al. (2022); Zheng et al. (2019); Wei et al. (2019b). AC algorithms, limited by data efficiency, require additional modules to enhance their performance in TSC problems (in limited simulation resources). DemoLight provides an excellent AC algorithm model, which biases the model towards superior control strategies from the beginning by pre-training the actor part with an expert model Xiong et al. (2019). UniTSA developed a traffic state enhancement method that enhances the model’s understanding of different intersections and allows for the fine-tuning of the pre-trained model at new intersections Wang et al. (2023). DenseLight proposed an unbiased dense reward named IFDG, enabling the model to obtain more effective data Lin et al. (2023). In addition to the above situations, there is also work that divides complex traffic problems into multiple sub-problems for solution Wei Huang and Lo (2023). Furthermore, a multi-view encoder for adaptive traffic signal control is proposed for better capture the environment, but such encoder requires a significant amount of 2D state information and is not data-efficient enough.

In TSC problems, the state space to be explored is often extremely large. In model-based RL, the exploration space can be minimized by computing state similarities, albeit at a high computational cost. Two novel algorithms have been introduced to approximate bisimulation metrics in deterministic MDPs Castro (2019). Additionally, an architecture is proposed to filter out task-irrelevant information from the state space Zhang et al. (2020). However, for TSC problems, such approaches either entails significant computational expenses or fails to achieve the desired outcomes.

### 3. METHOD

#### 3.1. *Problem Formulation*

The road network consists of  $m$ -intersections and can be defined as  $N = \{n_1, n_2, n_3, \dots, n_m\}$ , the structure of the intersection is shown in Figure 1, and an in-



**Figure 1.** Intersection structure

tersection  $n$  is taken from  $N$  as an example. Intersection  $n$  has two kinds of lanes: incoming lanes ( $l \in L_i$ ) and outgoing lanes ( $l \in L_o$ ). The two lanes are combined to form connectors. A connector  $c(l, m)$  means that vehicles can enter the intersection on lane  $l$  and leave on lane  $m$ . Associate a distinct vehicles number with each connector, and let  $x(l, m)$  be the number at the beginning of period  $t$ . Now, the ability to empty vehicles at a connector can be calculated by the following equation:

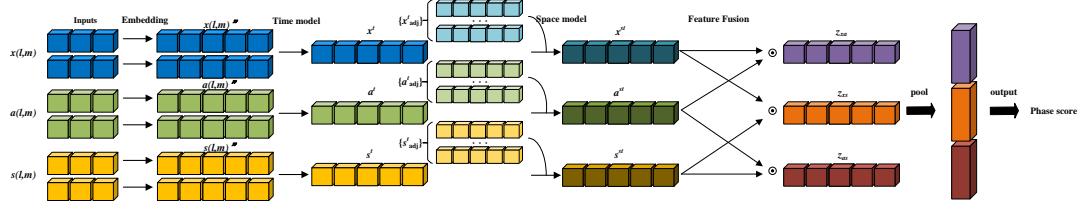
$$\begin{aligned} x(l, m)(t+1) &= x(l, m)(t) \\ &\quad - [C(l, m)(t+1) S(l, m)(t) \wedge x(l, m)(t)] \\ &\quad + d(l, m)(t+1), l \in L_i, m \in L_o \end{aligned} \quad (1)$$

$y \wedge z = \min(y, z)$  and  $S(l, m) = 1$  when connector  $c(l, m)$  is actuated otherwise 0,  $C(l, m)(t)$  is the number of the vehicles that can potentially depart by  $c(l, m)$ , and  $d(l, m)(t+1)$  is the sum of all arrivals from outside the current intersection, which can be calculated as:

$$\sum [C(k, l)(t+1) S(k, l)(t) \wedge x(k, l)(t)] R(l, m)(t+1) \quad (2)$$

For the scenario where a signal agent is shared globally,  $R(l, m)$  and  $S(l, m)$  are iid (independent, identically distributed). However, since there is only one agent, the scale of the road network is often limited due to the challenges in cooperation between adjacent agents. In the case of multiple agents,  $R(l, m)$  and  $S(l, m)$  are respectively controlled by different agents. Therefore, we can introduce a cooperative mechanism to achieve the goal of collaborative optimization. TSC problem is solved by finding an optimal strategy to control  $R(l, m)(t)$  and  $S(l, m)(t)$  to minimize the following expression:

$$\min \left( \sum_{t>0} \sum_{n \in N} \sum_{l \in L_i, m \in L_o} x(l, m)(t+1) - x(l, m)(t) \right) \quad (3)$$



**Figure 2.** The structure of critic

During the process of minimizing Equation 3, the throughput capacity of the entire road network for vehicles can be enhanced Varaiya (2013).

### 3.2. Overall Description of the Method

In our approach, we initially developed a state encoding method that accesses more pertinent data by analyzing the distribution of states accessed by various models. Building on this foundation, we integrated information from neighboring intersections and additional state information relevant to the task. This integration has strengthened the collaboration among agents and has more accurately captured the dynamic attributes of the traffic environment.

**Table 1.** Common notations

| Notation                    | Meaning                                    |
|-----------------------------|--------------------------------------------|
| $N$                         | Set of intersections                       |
| $n$                         | Single intersection                        |
| $l$                         | Set of imported lanes                      |
| $m$                         | Set of exit lanes                          |
| $c(l, m)$                   | Connector consisting of $l$ and $m$        |
| $x(l, m)$                   | Set of vehicles travelling from $l$ to $m$ |
| $S(l, m)$                   | Connector release sign                     |
| $\pi_{\theta_{\text{old}}}$ | Old agent's Strategies                     |
| $\pi_{\theta}(a_t   s_t)$   | Current agent's Strategies                 |
| $\mathcal{E}_1$             | Encoding Function                          |

### 3.3. Reinforcement Learning Design

In RL, the agent learns a behavioral policy  $\pi_{\theta}(a_t | s_t)$  through interactions with the environment, and maximize the reward function. A problem that can be addressed with RL is modeled as a Markov Decision Process (MDP), which is defined by a tuple  $\langle S, A, R, \gamma, P \rangle$ . Such tuple contains a finite set of states  $S$  and a finite set of actions  $A$ .  $P$  represents the transition function and can be shown as  $P : S \times A \times S \rightarrow [0, 1]$ . The reward function  $R$  can be defined as  $S \times A \times S \rightarrow R$  and  $\gamma$  is the reward discount factor. In an MDP,  $P$  is determined by the environment,  $\gamma$  can be adjusted as a parameter, and the state, action and reward functions are manually designed. In the process of solving the TSC problem, they are designed as following:

- (1) **State.** Given that the traffic signal at the current intersection controls only the vehicles on  $L_i$ , the information of these vehicles is collected as part of the state. While the model is attempting to minimize the value of Equation (1),  $x(l, m)$  is directly used as a component of the state. Furthermore, to prevent the model from misunderstanding the environment due to the information from a single pattern, we also bind the average speed  $s(l, m)$  and the average acceleration  $a(l, m)$  to connector  $c(l, m)$  and include them as part of the state. Thus, the entire state space is defined as  $\{x(l, m), a(l, m), s(l, m)\}_{l \in L_i, m \in L_o}$  and notice that at the same intersection,  $a(l, m)$  and  $s(l, m)$  are strongly correlated with  $x(l, m)$ , which will reduce the consumption of introducing additional state.
- (2) **Action.** The policy  $\pi_\theta(a_t | s_t)$  controls the traffic flow at the intersection by controlling the values of  $S(l, m)$  at time  $t$ , and  $S(l, m)$  has two options, where 1 denotes the connector  $c(l, m)$  is available for vehicles to pass and 0 otherwise. Here, a set  $C$  is predefined, containing multiple elements representing phases. Each phase consists of connectors without conflicts. Subsequently, the action is defined as selecting a phase from set  $C$  and setting  $S(l, m)$  values of the connectors composing the phase to 1. Furthermore, to accommodate pedestrian crossing requirements, the minimum duration of each phase is set to 10 seconds, followed by a 3-second yellow light interval to clear vehicles.
- (3) **Reward.** The objective of policy  $\pi_\theta(a_t | s_t)$  is to select an action based on the state to minimize Equation 3. We do not directly use Equation 3 as the reward function because it is too sparse, and as an alternative, the reward function is defined in accumulated waiting time as follows:

$$r(s_t, a_t) = w_t - w_{t+T} \quad (4)$$

$w_t$  represents the accumulate waiting time of vehicles on  $L_i$  at time  $t$ . The reward function is defined as the difference in accumulated waiting time before and after executing an action.

### 3.4. State Encoder

The Proximal Policy Optimization (PPO) algorithm update its policy network by constraining the difference between the old and new policies, and the loss can be formulated as:

$$\max_{\theta} \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} \left\{ \min \left[ r_t(\theta) \hat{A}(s_t, a_t), \right. \right. \quad (5)$$

$$\left. \left. \text{clip} (r_t(\theta), 1 - \epsilon, 1 + \epsilon) \cdot \hat{A}(s_t, a_t) \right] \right\}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \quad (6)$$

$$\hat{A}(s_t, a_t) = r_t + \gamma V(s_{t+1}) - V(s_t) \quad (7)$$

$r_t(\theta)$  represents the difference between new policy  $\pi_\theta(a_t | s_t)$  and the  $\pi_{\theta_{\text{old}}}$ , and  $\hat{A}(s_t, a_t)$  is to measure the reasonableness of performing action  $a_t$  at state  $s_t$  Schulman et al. (2017). By calculating the action  $a_t$  corresponding to the maximum, Equation 3 can be minimized. Back to TSC problems, considering the extremely large state space, it is difficult to access all states in limited simulation time, which will result in sub-optimal solutions.

A scalar describing the capacity of the road network is calculated at the end of the whole simulation according to Equation 3, but such a scalar is quite sparse for the whole traffic process. Therefore, by expanding the intermediate steps of Equation 3 and noting that  $x(x, m)(t)$  is a known constant to  $\pi_\theta(a_t | s_t)$ , the equation can be transformed into the following form:

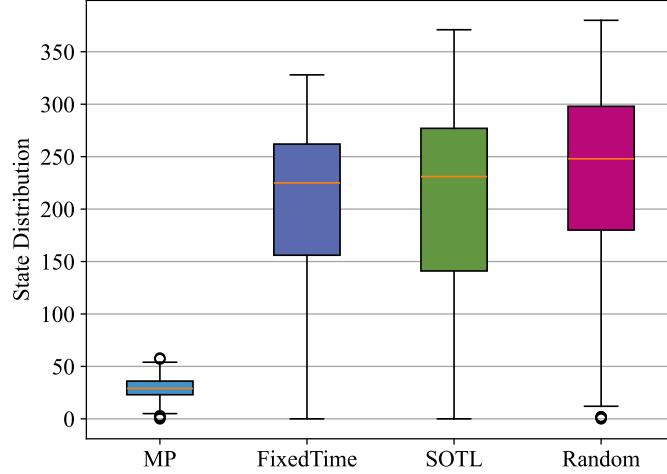
$$\sum_{t>0} \sum_{n \in N} \sum_{l \in L_i, m \in L_o} \text{argmin}(x(l, m)(t + 1)) \quad (8)$$

Now, Equation 3 is densified at time scale. At each time step after an action is executed, the value of  $x(l, m)(t + 1)$  can be calculated to assess the effectiveness of the current action. However, the model still struggles to access all states. So we had to decide which part of the state is more important to access. At this point, we statistic the distribution of states accessed by different control models and compare their performance.

**Table 2.** Performance of Different Control Models

| Method      | Travel Time | Throughput |
|-------------|-------------|------------|
| MaxPressure | 362.50      | 2728       |
| FixedTime   | 531.74      | 2425       |
| SOTL        | 541.50      | 2430       |
| Random      | 580.78      | 2418       |

After obtaining the performance of the two models, we conduct statistical analysis on the frequency of visits to different  $x(l, m)(t + 1)$  value by the two models. The



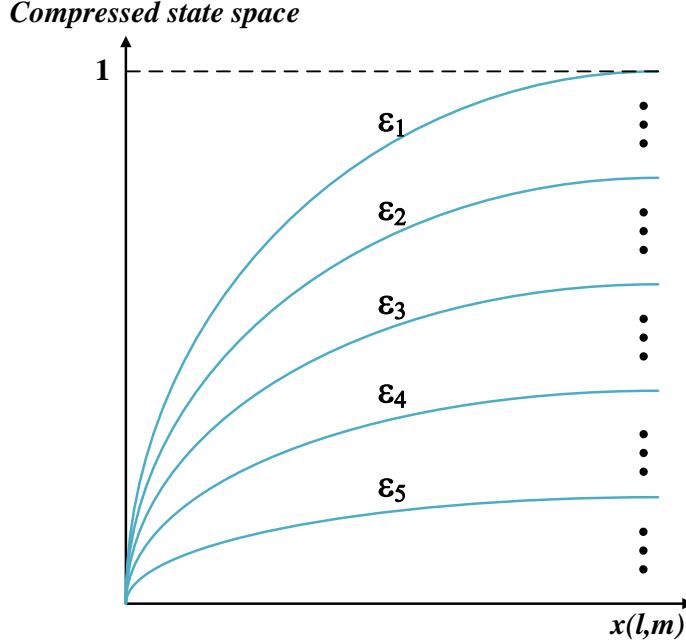
**Figure 3.** State distribution

performance of MaxPressure significantly surpasses that of the random model. This difference in performance is directly caused by the frequency of access to different  $x(l, m)(t + 1)$  values, and the complete model accesses states with low  $x(l, m)(t + 1)$  values (the optimization objective) significantly more frequently than the random

model. From the prior statistics of state access frequency, we got the basis for state encoding, where the model perfers the choice of state, specifically in the TSC problem,  $\pi_\theta(a_t \mid s_t)$  tends to choose a state with a lower value of  $x(l, m)(t+1)$ . For  $s_t = x(l, m)(t)$ , define an encoding function  $\mathcal{E}$ , whose effect is biased for different states. Based on the discussion above, the encoding function has the following properties:

- Compress the state space.
- Retain as much information as possible about the state that make  $x(l, m)(t+1)$  small.
- The compressed part of the state space is in the state region that makes  $x(l, m)(t+1)$  large.

The family of encoding functions is depicted in Figure 4: The vertical axis represents



**Figure 4.** Encoding function

the proportion of the normalized state space that is compressed.  $\mathcal{E}_1$  compresses all of the higher dimensional space, and  $\mathcal{E}_5$  retains as much high-dimensional information as possible. The intermediate encoding function strikes a balance between compressing the state space and preserving information. Ultimately, we employ logarithmic form for  $\mathcal{E}$  as follow:

$$\mathcal{E}(s) = [\lfloor (\log(x(l_1, m_1) + 1)) \rfloor \dots \lfloor (\log(x(l_n, m_n) + 1)) \rfloor] \quad (9)$$

where  $l_1, l_2 \dots l_n \in L_i$ ,  $m_1, m_2 \dots m_n \in L_o$ . Now, a compressed representation for state  $x(l, m)(t+1)$  has been obtained. Since the information of states with high  $x(l, m)(t+1)$  values are compressed, this may result in information loss. Therefore, in the construction of the spacetime model, additional state information is introduced to offset these losses. For the same intersection, different state information has potential connections. In the feature fusion module, this potential connection is captured to reduce additional exploration costs.

### 3.5. Spacetime Feature Model

To illustrate the process of the entire spacetime dependency model, take  $x(l, m)$  as an example ( $a(l, m), s(l, m)$  will be processed in the same manner). Since the encoding function  $\mathcal{E}$  takes effect before entering the network, for simplicity of expression,  $x(l, m)$  is still used here to represent the compressed state  $\mathcal{E}(x(l, m))$ .

At intersection  $i$  in time  $t$ ,  $\{x(l, m), a(l, m), s(l, m)\}$  is obtained. Processing the local state and the neighboring state in the same way and the state is updated by a two-layer Multi-layer Perceptron (MLP) Taud and Mas (2018).

$$x(l, m)'' = f_{c2}(f_{c1}(x(l, m))) \quad (10)$$

$f_{c1}$  and  $f_{c2}$  are one-layer MLP. The model embeds local and neighboring states into the same high-dimensional space.  $f_{c2}$  is a parameter-sharing layer among isomorphic intersections, which play a role in cooperation of different intersections.

After embedding the state, temporal dependency relies on historical information to be learned. Traffic flow states have strong temporal properties and we rely on such properties to address partially observable problems. Long Short-Term Memory unit is chosen to capture temporal dependencies Hochreiter and Schmidhuber (1997). This part of the update process is as follows:

$$f_t = \sigma(W_f [x_t'', x_{t-1}^{hid}] + b_f) \quad (11)$$

$$i_t = \sigma(W_i [x_t'', x_{t-1}^{hid}] + b_i) \quad (12)$$

$$\tilde{C}_t = \tanh(W_C [x_t'', x_{t-1}^{hid}] + b_C), \quad (13)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (14)$$

$$x^t = \sigma(W_o [O_t'', x_{t-1}^{hid}] + b_o), \quad (15)$$

$$x_t^{hid} = x^t \odot \tanh(C_t), \quad (16)$$

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

$$[x^t], (h_n, c_n) = LSTM(x(l, m)^{''t}) \quad (17)$$

$x(l, m)^{''t}$  is the temporal input sequence,  $h_n$  containing the initial hidden state for each element in the input sequence and  $c_n$  containing the initial cell state. The last element of  $h_n$  is the initial hidden state of current state. Write the state that has passed through the time dependency model as  $x^t$ , and the characteristics of adjacent intersections are represented as  $x_{adj}^t$ .

After extracting temporal dependencies, spatial dependencies that portray spatial relationships between different agents are noticed. The attention mechanism is applied to the network's updating process Velickovic et al. (2017). Unlike the common practice of feeding the entire road network structure into the network (such behavior is impossible for TSC problems with high real-time requirements), information from the agent with its first-order neighboring agents is aggregated. The features of the current intersection and neighboring intersections that have extracted timing information are

used to extract spatial dependencies. The process can be written as:

$$x^{st} = GAT(x^t, \{x_{adj}^t\}) \quad (18)$$

$\{x_{adj}^t\}$  is a set with information about corresponding neighboring nodes.  $x^{st}$  is the feature after gaining spatial dependencies and temporal dependencies. The detail update process can be represented as follows:

$$e_{ij} = a([Wx^t \| Wx^t]), j \in \mathcal{N}_i \quad (19)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(e_{ik}))} \quad (20)$$

$$x^{st} = \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij} Wx^t \right) \quad (21)$$

Then, we try to tap into the underlying relationships between states.

### 3.6. Feature Fusion

In RL, the state is used as an observation of the environment. A signal state is difficult to fully express the features of the environment, but a complex state can lead to state space explosion. In this paper, the state space is represented by  $\{x(l, m), a(l, m), s(l, m)\}_{l \in L_i, m \in L_o}$ , after feeding them into the spacetime model we get  $\{x_{st}, a_{st}, s_{st}\}$ . In subsequent reasoning, continuing to use the complete state information will make it difficult for the model to converge. Therefore, a network based on bilinear pooling (BP) is proposed that learns to capture the relationship between different features to obtain enhanced features Fukui et al. (2016). Mathematically, the feature fusion process can be described as follows:

$$\xi(\mathcal{I}) = \sum_l b(f_A, f_B) \quad (22)$$

$$x = \text{vec}(\xi(\mathcal{I})) \quad (23)$$

$$y = \text{sign}(x) \sqrt{|x|} \quad (24)$$

$$z = y / \|y\|_2 \quad (25)$$

$f_A$  and  $f_B$  represent different features of states, and the result  $z$  is the representation of fusion between two features. So, we get  $\{z_{xa}, z_{xs}, z_{as}\}$ , the element denotes features after fusion, and all the elements are stitched together as the final feature  $z_{xas}$ .  $z_{xas}$  is passed through the output layer to generate a normalised probability distribution (in Actor framework), the value of which indicates the probability that the corresponding action will be performed.

## 4. Experiments

### 4.1. Settings

Experiments are conducted on CityFlow<sup>1</sup>, an open-source, microscopic, multi-modal traffic simulation platform that provides interfaces for accessing traffic flow information and simulating realistic traffic scenarios Tang et al. (2019). Using the road network of Hangzhou and Jinan as datasets, traffic volumes at different time periods are collected on both networks to verify the generalization ability of the proposed model under different traffic conditions. Given that the effectiveness of RL at a single intersection has been well established in previous studies, the experiments focus on more complex scenarios and no longer include datasets for a single intersection.

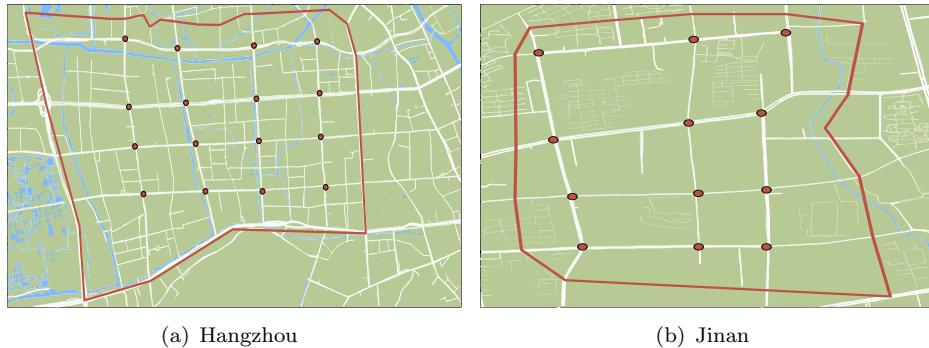
### 4.2. Evaluation Metrics

We choose travel time and total throughput as evaluation metrics. The calculation method for total throughput has been specifically described in the problem formulation, and for travel time, average all travel time of vehicles in the system. These two-evaluation metrics can better assess a model's ability to optimize traffic control Varaiya (2013); Wei et al. (2019a); Zhang et al. (2021). For travel time, the smaller the better, and for total throughput, the bigger the better.

### 4.3. Datasets

Our method is tested on real-world datasets<sup>2</sup> obtained from Jinan and Hangzhou, which is the most frequently used datasets to judge performance in the transportation field. The datasets are introduced as follows:

- $D_{Hangzhou}$ . A  $4 \times 4$  gird network. The traffic flow data are generated from roadside cameras, processed, and rewritten into CityFlow-conformant documents. Two sets of traffic data for the gird road network are obtained.
- $D_{Jinan}$ . Similar to  $D_{Hangzhou}$ , these traffic data are collected by roadside cameras near 12 intersections. The gird network is  $3 \times 4$  with each intersection connected to at least one other intersection.



**Figure 5.** Experimental road networks

Based on these datasets, we run the simulation for one hour of the real world, which

<sup>1</sup><http://cityflow-project.github.io>

<sup>2</sup>Datasets can be found at:<http://traffic-signal-control.github.io>

is equivalent to 3600 simulation steps. Simultaneously, traffic flows at different time points are collected to compare the model's performance at various time points within the same scenario. In the same set of experiments, the random seed is fixed.

#### 4.4. Compare Method

To demonstrate the superiority of our method, several state-of-the-art TSC methods are selected for comparison, including traditional signal control algorithms and RL-based methods (value-based and AC algorithms). Our model is denoted as  $E-ST-F$ , where  $E$  represents the encoding function,  $ST$  represents the spacetime feature model, and  $F$  represents the feature fusion module.

##### Traditional methods:

- Fixed-time Koonce et al. (2008): Summarize a fixed timing scheme from historical data. The phase order as well as the phase duration is fixed.
- MaxPressure Varaiya (2013): Activate the phase with the largest pressure each time. Pressure is the difference between the number of vehicles in the upstream queue and the downstream queue.

##### RL-based methods:

- MPLight Chen et al. (2020). Propose a deep reinforcement learning method to tackle the problem of city-level traffic signal control. The first to evaluate the RL-based traffic signal control methods in a real-world scenario with thousands of traffic lights.
- CoLight Wei et al. (2019b). A recent method using graph attention network for multi-intersection traffic light control. This method determines the agent's neighbors using rules and the number of each agent's neighbors is predefined.
- AttendLight Oroojlooy et al. (2020): An end-to-end framework for the problem of TSC. Training a single, universal model for intersections with any number of roads, lanes, phases (possible signals), and traffic flow.
- IPPO Schulman et al. (2017). As a baseline in ablation experiments on encoding functions.

#### 4.5. Result

Table summarizes the performance of all compared method, the best **boldfaced** and second best underline. The left side of the table compares the travel times of different methods, while the right side compares the throughput capacity of the road network.

Comparing traditional and RL-based methods reveals that the latter outperforms the former in terms of both travel time and throughput. The disparity in performance becomes even more pronounced under conditions of high traffic, highlighting the limitations of traditional methods in adapting to dynamic traffic environment.

A comparison of various RL-based methods on the Hangzhou road network demonstrates that our proposed model consistently outperforms existing models. With higher traffic volumes, our model achieves a 4.7% increase in throughput and an 8% reduction in travel time. Conversely, under conditions of lower traffic volumes, the model shows a modest throughput improvement of 1.2% and a travel time reduction of 9.8%. These results suggest that the optimization benefits of our model enhance as traffic volumes escalate.

Comparison of RL methods on Jinan road network. Jinan road network has collected

**Table 3.** Performance comparison of travel time.

| Method        | Jinan      |            |            | Hangzhou   |            |
|---------------|------------|------------|------------|------------|------------|
|               | Flow 1     | Flow 2     | Flow 3     | Flow 1     | Flow 2     |
| FixedTime     | 412        | 445        | 499        | 558        | 542        |
| MaxPressure   | 393        | 347        | 319        | 441        | 480        |
| MPLight       | 338        | 312        | 312        | 351        | 437        |
| AttendLight   | 332        | 300        | 301        | <u>335</u> | 446        |
| CoLight       | <u>316</u> | <u>297</u> | <b>296</b> | 342        | <u>427</u> |
| <b>E-ST-F</b> | <b>290</b> | <b>278</b> | <u>300</u> | <b>302</b> | <b>392</b> |

**Table 4.** Performance comparison of total throughput.

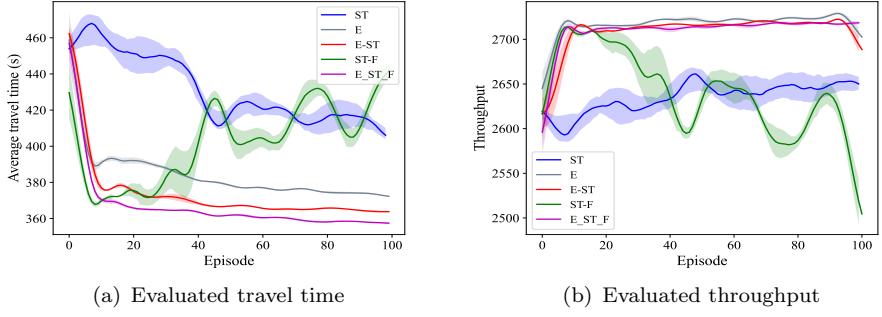
| Method        | Jinan       |             |             | Hangzhou    |             |
|---------------|-------------|-------------|-------------|-------------|-------------|
|               | Flow 1      | Flow 2      | Flow 3      | Flow 1      | Flow 2      |
| FixedTime     | 4431        | 3252        | 2948        | 1165        | 3943        |
| MaxPressure   | 4822        | 3763        | 3431        | 2043        | 4021        |
| MPLight       | 6005        | 5279        | 4272        | 2896        | 5054        |
| AttendLight   | 6006        | <u>5315</u> | 4279        | <u>2900</u> | 4958        |
| CoLight       | <u>6051</u> | 5283        | <b>4374</b> | 2898        | <u>5074</u> |
| <b>E-ST-F</b> | <b>6171</b> | <b>5404</b> | <u>4321</u> | <b>2935</b> | <b>5308</b> |

three types of traffic flow data; however, the impact of traffic volume on the Jinan road network is not as pronounced as on the Hangzhou road network. The travel times of vehicles are similar across the three traffic volumes. Our proposed model also demonstrates significant optimization effects on Jinan road network, with CoLight’s optimization effect being comparable to that of our proposed method under low traffic volume.

#### 4.6. Ablation Study

In this section, we investigate on how different components affect  $E - ST - F$ . Subsequently, additional experiments is conducted to explore the applicability range of the encoding function defined in this work. All experiments employ the same settings, and for models lacking spacetime model or feature fusion model, the state are concatenated.

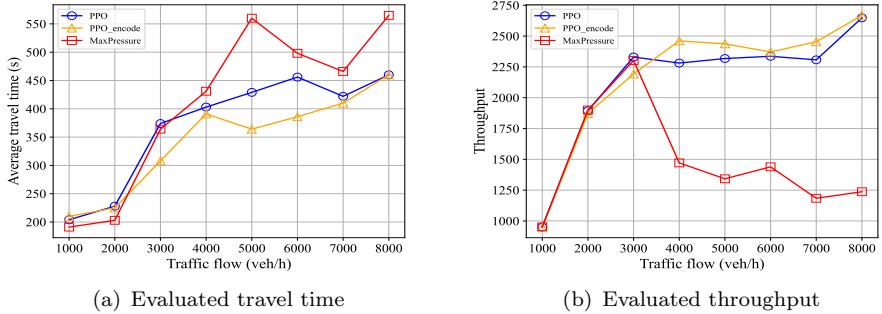
For models without encoding functions, their difficulty in converging is exacerbated due to the use of state spaces that combine multiple observations. Additionally, the encoding function ensures the stability of the model throughout the convergence process. The performance of models with encoding functions demonstrates the effectiveness of the employed spacetime model and feature fusion model. The synergy between these



**Figure 6.** Ablation study of  $E - ST - F$

two allows the model to reconstruct the environmental state as accurately as possible from limited observational information. Furthermore, experimental results demonstrate the effectiveness of the encoding function in compressing vast state spaces.

A single intersection is extracted from the Hangzhou road network and generate various traffic flows to validate the applicability range of the encoding function. For comparison with traditional approaches, only  $x(l, m)(t)$  is utilized as the state in this experiment.



**Figure 7.** Evaluated travel time and throughput in different traffic flow

Traditional methods perform similarly to RL-based methods when the hourly traffic flow is less than 3000. However, as the traffic flow continues to increase, it becomes difficult to control the intersection's traffic flow, leading to continuous deterioration. In terms of vehicle throughput at the intersection, the model with the encoding function exhibits only a slight advantage. However, when comparing travel times, the model with the encoding function shows a clear advantage, with the increase in traffic flow not significantly affecting travel time.

## 5. Discussion

In the above chapters, we define the signal control problem and give an optimisation approach. Compared to other reinforcement learning algorithms, this approach has demonstrated enhanced capabilities in improving traffic capacity. Its advantages arise partly from minimizing redundant exploratory space, and partly from integrating diverse informational inputs, which provides the model with a more holistic comprehension of the environment.

Through the encoding function employed, fine-grained data elements are aggregated into a coarse-grained format, enhancing the ratio of information that is more effectively utilized by the model. Unlike our prior RL algorithms, which predominantly relied on comprehensive state information, this approach addresses the challenge of accessing complete states in high-traffic scenarios.

As the model compresses and integrates various types of state information, it achieves enhanced perceptual capabilities. In the aspect of feature fusion, we hypothesize an inherent linkage among diverse state categories. To this end, we introduce a module specifically developed to elucidate these subtle interconnections. Exploring the potential relationships between different states may be a direction for breakthroughs in signal control problems.

In addition to the above mentioned, we illustrate other advantages of state compression algorithms. Traffic environments require a high degree of safety. Therefore, control algorithms are often trained in traffic simulation environments. But limited simulation resources can result in agents failing to explore the complete state space. Thus when the model is deployed, it is possible to encounter unexplored states, called out-of-distribution (OOD). After applying the state compression algorithm, the unexplored states may be compressed onto the explored states. Thus, state compression algorithms help to deal with the case of OOD.

## 6. Conclusion

In this study, we propose a RL-based model with multi-state fusion to enhance the model's understanding of the environment. This model mitigates the explosion of state space through state encoding, and facilitates cooperation between adjacent intelligent agents through the characterization of spacetime relationships. Specifically, we believe that the proposed encoding method can be applied to other control problems with large state spaces, despite variations in their specific state formulations.

We acknowledge the limitations of our model and would like to point out several future directions. In this study, the encoding function is formulated in logarithmic form; however, we believe that a more effective form of the encoding function can be derived from the specific state distribution of the complete model. In addition, we posit that once the issue of secure sampling is addressed, the proposed model can be effectively applied to real-world signal control challenges.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work was supported by the National Natural Science Foundation of China under Grant 71971116.

## 7. References

### References

- Cao, Da, Yawen Zeng, Meng Liu, Xiangnan He, Meng Wang, and Zheng Qin. 2020. “STRONG: Spatio-Temporal Reinforcement Learning for Cross-Modal Video Moment Localization.” *Proceedings of the 28th ACM International Conference on Multimedia* .
- Castro, Pablo Samuel. 2019. “Scalable methods for computing state similarity in deterministic Markov Decision Processes.” *ArXiv* abs/1911.09291.
- Chen, Chacha, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui. 2020. “Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control.” In *AAAI Conference on Artificial Intelligence* .
- Chen, Yiqun, Hangyu Mao, Tianle Zhang, Shiguang Wu, Bin Zhang, Jianye Hao, Dong Li, Bin Wang, and Hong Chang. 2022. “PTDE: Personalized Training with Distillated Execution for Multi-Agent Reinforcement Learning.” *ArXiv* abs/2210.08872.
- Chu, Tianshu, 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* 21: 1086–1095.
- Cools, Seung-Bae, Carlos Gershenson, and Bart D’Hooghe. 2006. “Self-organizing traffic lights: A realistic simulation.” *ArXiv* abs/nlin/0610040.
- Du, Simon Shaolei, Sham M. Kakade, Ruosong Wang, and Lin F. Yang. 2019. “Is a Good Representation Sufficient for Sample Efficient Reinforcement Learning?” *ArXiv* abs/1910.03016.
- Farebrother, Jesse, Marlos C. Machado, and Michael H. Bowling. 2018. “Generalization and Regularization in DQN.” *ArXiv* abs/1810.00123.
- Feng, Yuxiang, Yifan Zhao, Xingchen Zhang, Sérgio F. A. Batista, Yiannis Demiris, and Panagiotis Angeloudis. 2024. “Predicting spatio-temporal traffic flow: a comprehensive end-to-end approach from surveillance cameras.” *Transportmetrica B: Transport Dynamics* 12 (1). <https://doi.org/10.1080/21680566.2024.2380915>, <http://dx.doi.org/10.1080/21680566.2024.2380915>.
- Fukui, Akira, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. “Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding.” In *Conference on Empirical Methods in Natural Language Processing* .
- Gao, Juntao, Yulong Shen, Jia Liu, Minoru Ito, and Norio Shiratori. 2017. “Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network.” *ArXiv* abs/1705.02755.
- Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9: 1735–1780.
- Huang, Wei, Jing Hu, Guoyu Huang, and Hong K. Lo. 2023. “A three-layer hierarchical model-based approach for network-wide traffic signal control.” *Transportmetrica B: Transport Dynamics* 11 (1). <https://doi.org/10.1080/21680566.2023.2271174>, <http://dx.doi.org/10.1080/21680566.2023.2271174>.
- Koonce, Peter, Lee A. Rodegerdts, Kevin Lee, Shaun Quayle, Scott Beaird, Cade Braud, James A. Bonneson, Philip J. Tarnoff, and Thomas Urbanik. 2008. “Traffic signal timing manual.” .
- Liang, Enming, Z. C. Su, Chilin Fang, and Renxin Zhong. 2022. “OAM: An Option-Action Reinforcement Learning Framework for Universal Multi-Intersection Control.” In *AAAI Conference on Artificial Intelligence* .
- Lin, Junfan, Yuying Zhu, Lingbo Liu, Yang Liu, Guanbin Li, and Liang Lin. 2023. “Dense-Light: Efficient Control for Large-scale Traffic Signals with Dense Feedback.” *ArXiv* abs/2306.07553.
- Liu, Meiqi, J. Zhao, S. P. Hoogendoorn, and M. Wang. 2021. “An optimal control approach of integrating traffic signals and cooperative vehicle trajectories at intersections.” *Transportmetrica B: Transport Dynamics* 10 (1). <https://doi.org/10.1080/21680566.2021.1991505>, <http://dx.doi.org/10.1080/21680566.2021.1991505>.

- Lowrie, P. 1990. “SCATS: Sydney Co-Ordinated Adaptive Traffic System: a traffic responsive method of controlling urban traffic.” .
- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al. 2015. “Human-level control through deep reinforcement learning.” *Nature* 518: 529–533.
- Oroojlooy, Afshin, M. Nazari, Davood Hajinezhad, and Jorge Silva. 2020. “AttendLight: Universal Attention-Based Reinforcement Learning Model for Traffic Signal Control.” *ArXiv* abs/2010.05772.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. “Proximal Policy Optimization Algorithms.” *ArXiv* abs/1707.06347.
- Tang, Zheng, Milind R. Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, D. Anastasiu, and Jenq-Neng Hwang. 2019. “CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification.” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 8789–8798.
- Taud, Hind, and Jean-François Mas. 2018. “Multilayer Perceptron (MLP).” <https://api.semanticscholar.org/CorpusID:67464997>.
- Varaiya, Pravin Pratap. 2013. “Max pressure control of a network of signalized intersections.” *Transportation Research Part C-emerging Technologies* 36: 177–195.
- Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio’, and Yoshua Bengio. 2017. “Graph Attention Networks.” *ArXiv* abs/1710.10903.
- Wang, Maonan, Xi Xiong, Yuheng Kan, Chengcheng Xu, and Man-On Pun. 2023. “UniTSA: A Universal Reinforcement Learning Framework for V2X Traffic Signal Control.” *ArXiv* abs/2312.05090.
- Webster, Frank V. 1958. “TRAFFIC SIGNAL SETTINGS.” .
- Wei, Hua, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash V. Gayah, Kai Xu, and Zhenhui Jessie Li. 2019a. “PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network.” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* .
- Wei, Hua, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Jessie Li. 2019b. “CoLight: Learning Network-level Cooperation for Traffic Signal Control.” *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* .
- Wei, Hua, and Guanjie Zheng. 2021. “Recent Advances in Reinforcement Learning for Traffic Signal Control.” *ACM SIGKDD Explorations Newsletter* 22: 12 – 18.
- Wei, Hua, Guanjie Zheng, Vikash V. Gayah, and Zhenhui Jessie Li. 2019c. “A Survey on Traffic Signal Control Methods.” *ArXiv* abs/1904.08117.
- Wei Huang, Guoyu Huang, Jing Hu, and Hong K. Lo. 2023. “A three-layer hierarchical model-based approach for network-wide traffic signal control.” *Transportmetrica B: Transport Dynamics* 11 (1): 1912–1942. <https://doi.org/10.1080/21680566.2023.2271174>.
- Xiong, Yuanhao, Guanjie Zheng, Kai Xu, and Zhenhui Jessie Li. 2019. “Learning Traffic Signal Control from Demonstrations.” *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* .
- Zhang, Amy, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. 2020. “Learning Invariant Representations for Reinforcement Learning without Reconstruction.” *ArXiv* abs/2006.10742.
- Zhang, Liang, Qiang Wu, Jun Shen, Linyuan Lu, Bo Du, and Jianqing Wu. 2021. “Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control.” In *International Conference on Machine Learning*, .
- Zheng, Guanjie, Yuanhao Xiong, Xinshi Zang, J. Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Jessie Li. 2019. “Learning Phase Competition for Traffic Signal Control.” *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* .