Learning Phase Competition for Traffic Signal Control

Guanjie Zheng[†], Yuanhao Xiong[‡], Xinshi Zang[§], Jie Feng[¶], Hua Wei[†]
Huichu Zhang[§], Yong Li[¶], Kai Xu[ᵀ], Zhenhui Li[†]
[†]The Pennsylvania State University, [‡]Zhejiang Univerisity, [§]Shanghai Jiao Tong Univerisity,
[¶]Tsinghua Univerisity, [ᵀ]Shanghai Tianrang Intelligent Technology Co., Ltd
[†]{gjz5038, hzw77, jessieli}@ist.psu.edu, [‡]xiongyh@zju.edu.cn, [§]zang-xs@foxmail.com, [¶]feng-j16@mails.tsinghua.edu.cn,
[§]zhc@apex.sjtu.edu.cn, [¶]liyong07@tsinghua.edu.cn, [ᵀ]kai.xu@tianrang-inc.com

ABSTRACT

Increasingly available city data and advanced learning techniques have empowered people to improve the efficiency of our city functions. Among them, improving urban transportation efficiency is one of the most prominent topics. Recent studies have proposed to use reinforcement learning (RL) for traffic signal control. Different from traditional transportation approaches which rely heavily on prior knowledge, RL can learn directly from the feedback. However, without a careful model design, existing RL methods typically take a long time to converge and the learned models may fail to adapt to new scenarios. For example, a model trained well for morning traffic may not work for the afternoon traffic because the traffic flow could be reversed, resulting in very different state representation.

In this paper, we propose a novel design called FRAP, which is based on the intuitive principle of phase competition in traffic signal control: when two traffic signals conflict, priority should be given to one with larger traffic movement (i.e., higher demand). Through the phase competition modeling, our model achieves invariance to symmetrical cases such as flipping and rotation in traffic flow. By conducting comprehensive experiments, we demonstrate that our model finds better solutions than existing RL methods in the complicated all-phase selection problem, converges much faster during training, and achieves superior generalizability for different road structures and traffic conditions.

ACM Reference Format:
Guanjie Zheng[†], Yuanhao Xiong[‡], Xinshi Zang[§], Jie Feng[¶], Hua Wei[†] and Huichu Zhang[§], Yong Li[¶], Kai Xu[ᵀ], Zhenhui Li[†]. 2019. Learning Phase Competition for Traffic Signal Control. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357900

1 INTRODUCTION

Traffic congestion is one of the most severe urban issues today, which has resulted in tremendous economic cost and a waste of people's time. Congestion is caused by many factors, such as overloaded traffic and bad design of road structures. Some factors may require more sophisticated policy or long-term planning. But one direct factor that could be potentially improved by today's big data and advanced learning technology is traffic signal control.

Nowadays, the most widely used traffic signal control systems such as SCATS [21] and SCOOT [17] are still based on manually designed traffic signal plans. These plans, however, are not adaptive enough to the dynamics of today's complex traffic flows.

Recently, reinforcement learning (RL) has emerged as a promising solution to traffic signal control in real-world scenarios [37]. Unlike previous methods which rely on manually designed plans or pre-defined traffic flow models, RL methods directly learn the policy by interacting with the environment. To this end, a typical approach is to model each intersection as an agent and the agent optimizes its reward (e.g., travel time) based on the feedback received from the environment after it takes an action (i.e., setting the traffic signals). These RL approaches have shown promising results under simple traffic signal control settings, i.e., an intersection with a small number of signal phases, such as 2 or 4.

With more complex scenarios, learning the optimal policy becomes substantially much more difficult. Consider a standard four-approach intersection where each approach has left-turn, through and right-turn traffic. There will be 8 phases (i.e., combinations of different traffic movements) according to the traffic rules (see Section 3 for details). It turns out that it is much harder for the RL algorithm to deal with the 8-phase setting than the 2-phase setting. A close examination of the problem reveals that the difficulty is mainly due to the explosion of state space. In the 2-phase setting, there are only four through lanes. Assuming the state definition consists of the number of vehicles on each lane and current signal phase, and the vehicle capacity of a lane is n, the state space size of 2-phase control problem will be $2 \times n^4$ (enumerating the number of vehicles on each lane, under each phase). When all eight phases are considered, four extra left-turn lanes are added and the exploration space will increase to $8 \times n^8$. Therefore, the key challenge becomes how to reduce the problem space and explore different scenarios more efficiently, so that the RL algorithm can find the optimal solution within a minimal number of trials.

Surprisingly, none of the existing studies has attempted to address this issue. In fact, current RL methods are all exploring "blindly", wasting time on repeated situations. It is known that the principle of deep Q-network (DQN) is to use deep neural networks to approximate the state-action value $Q(s, a)$ and choose the action with the best value. Merely using convolution layers and fully-connected

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357900

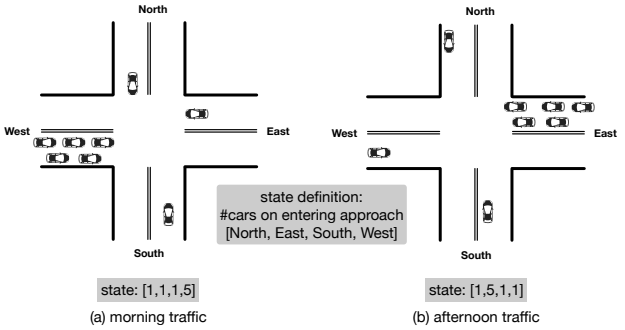


Figure 1: Traffic (a) and (b) are approximately flipped cases of each other.

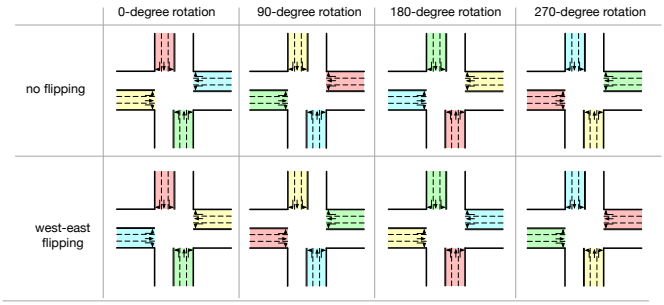


Figure 2: All the variations based on rotation and flipping of the left-most case. Ideally, a RL model should handle all these cases equally well.

layers, previous RL methods [32, 38] regress the $Q(s, a)$ value from the 8-lane input independently, i.e., since they have to try eight kinds of actions on each state, roughly $(8 \times n^8) \times 8$ samples will be needed for satisfactory approximation. But in fact, a considerable portion of state-action pairs are unnecessary to explore. Take Figure 1 for example. These two scenarios are approximately mirrored images (the traffic is flipped). Such a flipping is a common phenomenon when people commute from residential areas to commercial areas in the morning and travel the reverse way in the afternoon. Since such a flipping will result in a totally different state representation for existing methods, a RL agent which has learned the first case still cannot handle the second case. But based on the common sense, these two cases are almost identical and one would hope that the model learned from the first case can handle the second case or other similar cases. Furthermore, as shown in Figure 2, given any particular state, one can generate seven other cases through rotation and flipping. An ideal RL model is thus expected to handle all eight cases even only one case is seen during training.

Based on the observation above, we propose a novel model design called FRAP, which is invariant to symmetric operations like **F**lipping and **R**otation and considers **A**ll **P**hase configurations. The key idea is that, instead of considering individual traffic movements, one should focus on the relation between different traffic movements. This idea is based on the intuitive principle of competition in traffic signal control: (1) larger traffic movement indicates a higher demand for green signal; and (2) when two traffic movements conflict, we should give the priority to the one with higher demand.

Inspired by this principle, FRAP first predicts the demand for each signal phase, and then models the competition between phases. Through the pair-wise phase competition modeling, FRAP can achieve invariance to symmetries in traffic signal control (e.g., flipping and rotation). By leveraging such invariance and enabling knowledge sharing across the symmetric states, FRAP successfully reduces the exploration space to $16 \times n^4$ samples from $64 \times n^8$ (see Section 4 for detailed analysis). Compared to existing RL-based methods, FRAP finds better policies and converges faster under complex scenarios. Note that, our model design is independent of recent advances in reinforcement learning techniques, including different algorithms like value-based methods [16], and policy-based methods [26], and network design or training tricks like Double DQN [33], Dueling network [34]. Therefore, our method can be combined with these algorithms or tricks to achieve better performance. However, these are not the scope of this paper. Hence, to focus on validating the effectiveness of our design, we use a recent state-of-the-art Ape-X DQN [16] as a base for our model design.

In summary, the main contributions of this paper include:

- We propose a novel model design FRAP for RL-based traffic signal control. By capturing the competition relation between different signal phases, FRAP achieves invariance to symmetry properties, which in turn leads to better solutions for the difficult all-phase traffic signal control problem.
- We demonstrate that FRAP converges much faster than existing RL methods during the learning process through comprehensive experiments on real-world data.
- We further demonstrate the superior generalizability of FRAP. Specifically, we show that FRAP can handle different road structures, different traffic flows, complex real-world phase settings, as well as a multi-intersection environment.

2 RELATED WORK

Traditional traffic signal control. Traffic signal control is a core research topic in the transportation field and existing methods can be generally categorized into four classes.

Fixed-timed control [29] decides a traffic signal plan according to human prior knowledge and the signal timing does not change according to the real-time data.

Actuated methods [15, 24] define a set of rules and the traffic signal is triggered according to the pre-defined rules and real-time data. An example rule can be, to set the green signal for that traffic movement if the queue length is longer than a certain threshold.

Selection-based adaptive control methods first decide a set of traffic signal plans and choose the best one for the current traffic situation (e.g., traffic volume data received from loop sensors). This method is widely deployed in today's traffic signal control. Commonly used systems include SCATS [21], RHODES [25] and SCOOT [17].

All the methods mentioned above highly rely on human knowledge, as they require manually designed traffic signal plans or rules.

Optimization-based adaptive control approaches rely less on human knowledge and decide the traffic signal plans according to the observed data. These approaches typically formulate traffic signal control as an optimization problem under certain traffic flow models.

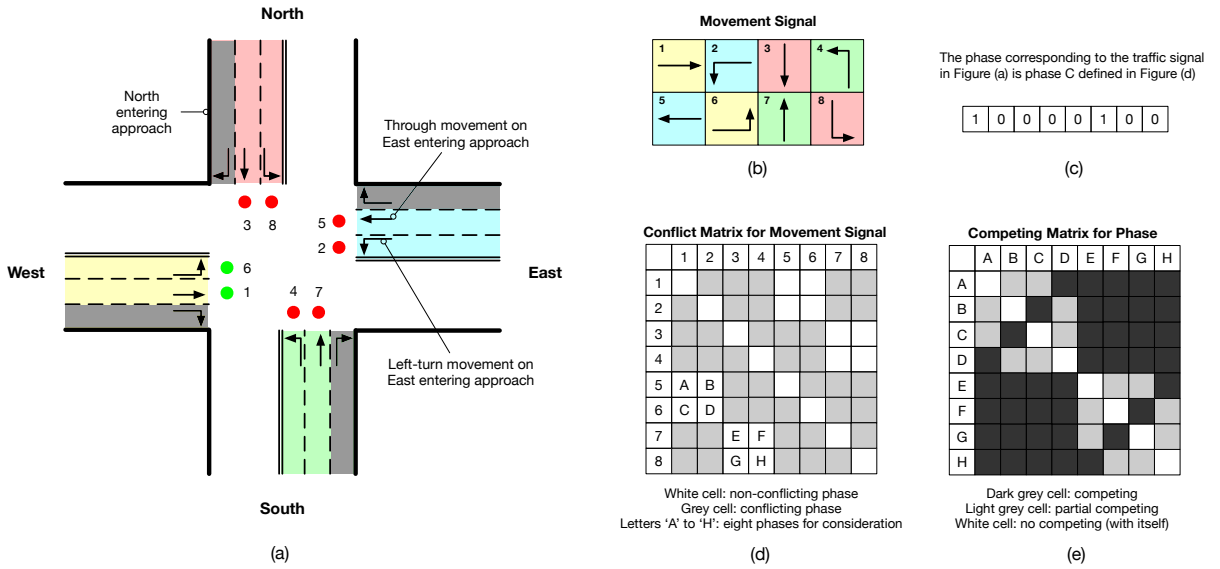


Figure 3: Illustration of preliminary definition.

To make the optimization problems tractable, strong assumptions about the model are often made. For example, a classical approach is to optimize travel time by assuming uniform arrival rate [29, 35]. The traffic signal plan including cycle length and phase ratio can then be calculated using a formula based on the traffic data. However, the model assumptions (e.g., uniform arrival rate [29, 35]) are often too restricted and do not apply in the real world.

Learning for traffic signal control. Different from traditional methods, learning-based traffic signal control does not require predefined traffic signal plan or traffic flow models. In particular, RL methods directly learns from intersections with the world. In these methods, each intersection is an agent, state is the quantitative description of the traffic condition at that intersection, action is the traffic signal, and reward is a measure of transportation efficiency.

Existing RL methods differ in terms of state description of the environment (e.g., image of vehicle positions [6, 18, 32, 38, 40], queue length [1–3, 38, 40], waiting time [7, 28, 38, 39, 41]), action definition (e.g., change to next phase [7, 28, 32, 38], setting a phase [1, 2, 8, 30]), and reward design (e.g., queue length [6, 22, 32, 36], delay [7, 12, 13, 32]). In terms of algorithms, studies have utilized tabular methods (e.g., Q-learning [3, 12]) for discrete state space and approximation methods [19, 38], which can be further categorized into value based (e.g., deep Q-Network [19, 32, 38]), policy based (e.g., policy gradient [27]), and actor critic [4, 5, 9].

However, to the best of our knowledge, none of these methods have shown satisfactory results in the complete 8-phase scenario for one single intersection due to the large exploration space. In this paper, we follow the universal principles of competition and invariance in traffic signal control to design a novel model for efficient exploration. Further, we adopt the distributed framework of Ape-X DQN [16] as our base framework, which is shown to achieve state-of-the-art performance in playing Atari games. But our model design can be adapted to other algorithmic frameworks including policy-based and actor-critic based RL methods.

3 PROBLEM DEFINITION

3.1 Preliminary

In this paper, we investigate the traffic signal control in both single and multi-intersection scenarios. To illustrate the definitions, we use the 4-approach intersection shown in Figure 3 as an example. But the concepts can be easily generalized to different intersection structures (e.g., different number of entering approaches).

- **Entering approach**: Each intersection has four entering approaches, named as North / South / West / East entering approach ('N', 'S', 'W', 'E' for short) respectively. In Figure 3(a), we point out the North entering approach.
- **Traffic movement**: A traffic movement is defined as the traffic moving towards a certain direction, i.e., left turn, through, and right turn. In Figure 3(a), we show that there are 8 traffic movements controlled by the signal (right-turn not included). Follow the traffic rules in most countries, right turn traffic can pass regardless of the signal, but it needs to yield on a red light. In addition, a traffic movement could occupy more than one lane but this does not affect our model design because a traffic signal controls a traffic movement instead of a lane.
- **Movement signal**: For each traffic movement, we can use one bit with 1 as 'green' signal and 0 as 'red'.
- **Phase**: We use an 8-bit vector $\mathbf{p}$ to represent a combination of movement signals (i.e., a phase), as shown in Figure 3(c). As indicated by the conflict matrix in Figure 3(d), some signals cannot turn 'green' at the same time (e.g., signals #1 and #2). All the non-conflicting signals will generate 8 valid paired-signal phases (letters 'A' to 'H' in Figure 3(d)) and 8 single-signal phases (the diagonal cells in the conflict matrix). Here we do not consider the single-signal phase because, in an isolated intersection, it is always more efficient to use paired-signal phases.[1]

[1]When considering multiple intersections, a single-signal phase might be necessary because of the potential spill back.

Speaking of the relation between phase (Figure 3(e)), there are in total two categories: competing, and partial competing. Partial competing phases (e.g., 'A' and 'B') have one traffic movement in common, while competing phases do not. They should be treated differently in modeling (detailed in Section 4).

3.2 RL Environment

Driven by the idea of learning from the feedback, in this paper we propose a reinforcement learning approach to traffic signal control. In our problem, an agent can observe the traffic situation at an isolated intersection (Figure 3(a)) and change the traffic signals accordingly. The goal of the agent is to learn a policy for operating the signals which optimizes travel time. This traffic signal control problem can be formulated as a Markov Decision Process $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ [31]:

Problem 1. *Given the state observations set $\mathcal{S}$, action set $\mathcal{A}$, the reward function $\mathcal{R}$ is a function of $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, specifically, $\mathcal{R}_s^a = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$. The agent aims to learn a policy $\pi(A_t = a|S_t = s)$, which determines the best action a to take given state s, so that the following total discounted return is maximized:*[2]

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{m=0}^{\infty} \gamma^m R_{t+m+1}. \quad (1)$$

For traffic signal control, our RL agent is defined as follows:

- **State**: the number of vehicles $\mathbf{f}_i^v$ on each traffic movement i and current traffic signal phase (represented as one bit $\mathbf{f}_i^s$ for each traffic movement signal). Upper script v and s stand for vehicle and signal respectively.
- **Action**: to choose the phase for the next time interval.
- **Reward**: the average queue length of each traffic movement.

Note that, we use a relatively simple set of state features and reward function, for the reason that we focus on innovating the model design in this paper. Optimizing queue length has been proved to be equivalent to optimizing travel time (most widely accepted measurement in transportation) in [43]. In addition, the concise state design has also been shown effective [43]. Meanwhile, our method can easily incorporate more complex state features and rewards for performance boosts.

4 METHOD

4.1 Model Overview

We use Ape-X Deep Q-learning (DQN) to solve the RL problem. The network takes the state features on the traffic movements as input and predicts the score (i.e., Q value) for each action (i.e., phase) as described in the Bellman Equation [31]:

$$Q(s_t, a_t) = R(s_t, a_t) + \gamma \max Q(s_{t+1}, a_{t+1}). \quad (2)$$

The action with the highest score will be selected for this step.

We design a novel network called FRAP (which is invariant to symmetric operations like **F**lip and **R**otation and considers **A**ll **P**hase configurations) based on two universal principles:

- **Principle of competition**: Larger traffic movement indicates higher demand for 'green' movement signal. When two signals conflict, priority should be given to the one with higher demand.
- **Principle of invariance**: Signal control should be invariant to symmetries such as rotation and flipping.

This way, the learning for different traffic movements and phases can now occur at the same time by updating the same network module (i.e., parameters), which leads to more efficient use of the data samples and better performance.

The rest of this section is organized as follows. In Section 4.2, we give a brief overview of the state-of-the-art Ape-X DQN [16] framework, upon which our method is built. Then, we describe our network design in detail in Section 4.3. In Section 4.4, we further discuss some important properties of our model.

4.2 Algorithmic Framework

To improve the learning efficiency in large search space, we adopt the distributed framework Ape-X DQN [16] as our algorithmic framework. In Ape-X DQN, the standard deep reinforcement learning is decomposed into two parts: *acting* and *learning*. The *acting* part assigns multiple actors with different exploration policies to interact with an environment and to store the observed data in a replay memory. The *learning* part is responsible for sampling the training data in the replay memory to update the model. Most importantly, the two parts can run concurrently while keeping the speed of generating and consuming the training data almost equal. In short, benefiting from the high exploration and sampling efficiency, this framework can significantly boost the learning performance of reinforcement learning. For more details about Ape-X DQN, we refer interested readers to [16].

4.3 Phase Invariant Signal Control Design

As we discussed before, training a RL agent for traffic signal control is highly challenging due to the large search space. For instance, for the four-approach intersection shown in Figure 3(a), assuming there is only one lane on each traffic movement, the size of the state space will be $8 \times n^8$, where n is the capacity of a lane. Thus, even with a small lane capacity (e.g., $n = 10$), DQN will require billions of data samples to learn the relation between state, action and reward. Further, intersections may vary in the geometry (e.g., 3, 4, or 5 entering approaches) and the signal setting (i.e., a different combination of traffic movement signals). It is very inefficient if a different agent needs to be learned for each different intersection.

To address these challenges, we design our model based on the two principles outlined in Section 4.1, so that it can learn more efficiently from data and also be easily adapted to different intersection structures. We divide the prediction of phase score (i.e., Q value) into three stages: *phase demand modeling*, *phase pair representation*, and *phase pair competition*. Figure 4 shows an overview of our method, and Figure 5 shows the detail network parameters settings. Next, we describe the design in detail.

4.3.1 Phase Demand Modeling. In this stage, our goal is to obtain a representation of the demand for each signal phase. Recall that for any traffic movement $i, i \in \{1, \ldots, 8\}$, its state includes the number of vehicles and the current signal phase. These features can be obtained directly from the simulator. We first take these

[2]State transition probability matrix $\mathcal{P}$ is not described here because it is not explicitly modeled in model-free methods.

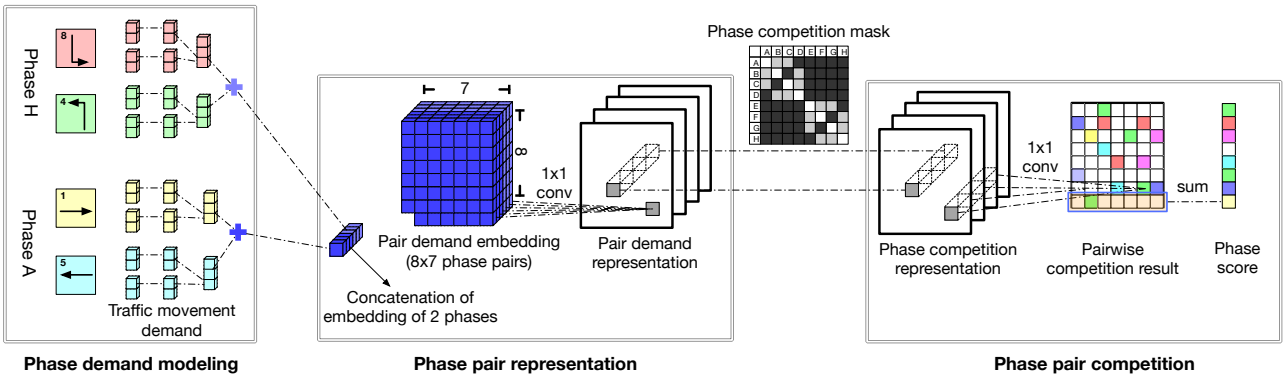


Figure 4: Network design of FRAP signal control.

two features, denoted as $\mathbf{f}_i^v$ and $\mathbf{f}_i^s$ respectively, as input, and pass them through a neural network of two fully-connected layers to generate a representation of the demand for 'green' signal on this traffic movement, $\mathbf{d}_i$.

$$\mathbf{h}_i^v = ReLU(\mathbf{W}^v \mathbf{f}_i^v + \mathbf{b}^v), \quad \mathbf{h}_i^s = ReLU(\mathbf{W}^s \mathbf{f}_i^s + \mathbf{b}^s). \tag{3}$$

$$\mathbf{d}_i = ReLU(\mathbf{W}^h [\mathbf{h}_i^v, \mathbf{h}_i^s] + \mathbf{b}^h). \tag{4}$$

Note that the two hidden layer vectors $\mathbf{h}_i^v, \mathbf{h}_i^s$ are combined before passed through the output layer. Further, the learned parameters of the neural network are shared among all traffic movements.

Finally, we obtain the demand representation of any phase $\mathbf{p}$ by adding together the demands of the two non-conflicting traffic movement signals in $\mathbf{p}$:

$$\mathbf{d}(\mathbf{p}) = \mathbf{d}_i + \mathbf{d}_j, \text{ where } \mathbf{p}_i = \mathbf{p}_j = 1. \tag{5}$$

4.3.2 Phase Pair Representation. By the principle of competition, the score (priority) of a phase depends on its competition with the other phases. Thus, for each phase $\mathbf{p}$, we form phase pairs $(\mathbf{p}, \mathbf{q})$ where $\mathbf{q}$ is an opponent of $\mathbf{p}$ (i.e., $\mathbf{q} \neq \mathbf{p}$). Given a pair $(\mathbf{p}, \mathbf{q})$ and their demands, our goal of this stage is therefore to obtain a representation of the competition between $\mathbf{p}$ and $\mathbf{q}$.

The pair demand embedding volume $\mathbf{D}$ (blue volume in Figure 4) is formed by first concatenating the demand representations of phases in a phase pair (i.e., $[\mathbf{d}(\mathbf{p}), \mathbf{d}(\mathbf{q})]$), and then gathering the vectors of all phase pairs. For our 8-phase problem, the size of $\mathbf{D}$ is $8 \times 7 \times l_1$, where l_1 is the length of the demand embedding vector for a single pair.

Then, we further apply K convolutional layers with 1×1 filters on $\mathbf{D}$ and obtain the phase pair demand representation. The k-th layer can be written as Eq. (6), with $\mathbf{H}_0^d = \mathbf{D}$.

$$\mathbf{H}_k^d = \text{ReLU}(\mathbf{W}_k^d \cdot \mathbf{H}_{k-1}^r + \mathbf{b}_k^d) \tag{6}$$

The choice of 1×1 filters follows the idea of extracting competing relationship in the phase pair by letting them interact with each other, which is verified in prior work [20]. 1×1 filter also enables the parameter sharing among different phase pairs. Because there is no explicit meaningful interaction between different phase pairs (e.g., 2, 3, 4, or more phase pairs), it is not useful to use larger filters.

4.3.3 Phase competition mask. For better understanding, this part is represented by the grey matrix in Figure 4, but is detailed in Figure 5. As illustrated in Figure 3(e), a phase pair $\mathbf{p}$ and $\mathbf{q}$ can have two different relations: partial competing (light grey, e.g., phase A and B, which shares one traffic movement) and competing (dark grey, e.g., phase A and D, which totally conflict with each other). Once the phase pair is determined, our model will look up the phase competing matrix and map the relation to an embedding vector $\mathbf{e}(\mathbf{p}, \mathbf{q})$. Putting together the embedding vectors of all the phase pairs forms the relation embedding volume $\mathbf{E}$. Similar to the pair demand representation modeling, K convolutional layers with 1×1 filters are applied on $\mathbf{E}$ to obtain the pair relation masking. The k-th layer can be written as Eq. (7), with $\mathbf{H}_0^r = \mathbf{E}$.

$$\mathbf{H}_k^r = \text{ReLU}(\mathbf{W}_k^r \cdot \mathbf{H}_{k-1}^r + \mathbf{b}_k^r) \tag{7}$$

4.3.4 Phase Pair Competition. In this stage, our model takes the phase pair demand representation and phase competition mask as input and predicts the score (i.e., Q-value) of each phase considering its competition with other phases.

Following the last stage, a phase competition representation $\mathbf{H}^c$ can be obtained by an element-wise multiplication of the phase pair demand representation $\mathbf{H}_K^d$ and the phase competition mask $\mathbf{H}_K^r$: $\mathbf{H}^c = \mathbf{H}_K^d \otimes \mathbf{H}_K^r$. We then apply another convolutional layer with 1×1 filter to get the pairwise competition result matrix $\mathbf{C}$, each row of which represents the relative priorities of a phase $\mathbf{p}$ over all its opponents. Mathematically, we have

$$\mathbf{C} = \text{ReLU}(\mathbf{W}^c \cdot \mathbf{H}^c + \mathbf{b}^c). \tag{8}$$

Finally, the relative priorities of each phase $\mathbf{p}$ are added together to obtain the score of phase $\mathbf{p}$. Our RL agent then chooses the phase with the highest score as its action.

4.3.5 Summary of FRAP network. In summary, we give a detailed description of FRAP network in Figure 5, which is consistent with the model design in Figure 4. The network is mainly made up of embedding layers and convolution layers. In the convolution layers, the filter is a 1×1 convolution with a stride of 1. By default, we use 20 filters in each convolution layer, except the last one.

4.4 Discussions

Invariance of our model design. Throughout the above modeling process, no matter which phase $\mathbf{p}$ we are focusing on, we always have a symmetric view of its relationship with other phases. This enables FRAP to leverage the symmetry properties in traffic signal control and greatly reduce the exploration of samples. Specifically, assume that a maximum of n vehicles is allowed on each movement. Note that, the traffic movement signal could be either '1' (green) or

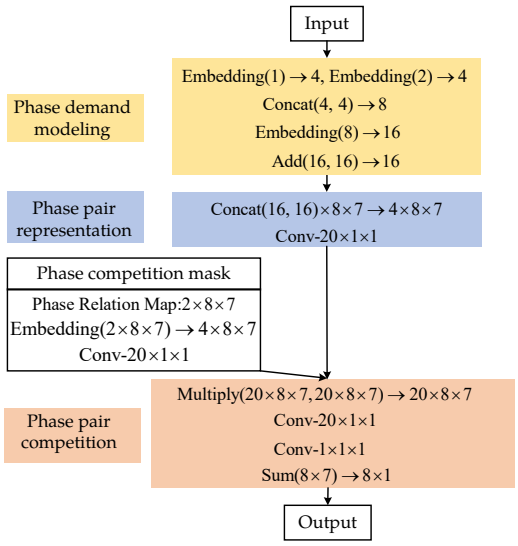


Figure 5: Network detail of FRAP.

Figure 6: Real-world 3-, 4-, and 5-approach intersections.

'0' (red). As shown in Figure 4, our network first obtains a representation for a phase from features of two traffic movements with $2^2 \times n^2$ possible combinations. Then two phases are paired together to compete and the model of phase competition is shared among all pairs. In this way, to regress Q values for all eight actions, the model is required to observe only $(2^2 \times n^2) \times (2^2 \times n^2) = 16 \times n^4$ samples, a significant decrease in comparison with $64 \times n^8$ as in DRL [32] and IntelliLight [38]. In Section 5, we further conduct extensive experiments to illustrate that FRAP converges faster and to better solutions, as the enhanced sample efficiency compensates for the explosion of state space.

Adaption to different environments. As we focus on the universal principles of phase competition and invariance throughout our model design, the FRAP model can be applied to different traffic conditions (i.e., small, medium and large traffic volumes), different traffic signal settings (e.g., 4-phase, 8-phase), and different road structures (e.g., 3-, 4-, and 5-approach intersections as shown in Figure 6, and intersections with a variable number of lanes on each traffic movement). Further, FRAP can learn from one environment and transfer to another one with high accuracy without any additional training. We further illustrate this in the experiments.

Applications in multi-intersection environments. Though our discussion so far has been only focused on single intersections, FRAP makes fundamental contributions to city-wide traffic signal control, as a good learning model at single intersection is the base unit even in the scale of city-wide traffic signal control. In addition, we demonstrate that FRAP works well in the multi-intersection environment even without explicit coordination (see experiments).

5 EXPERIMENT

5.1 Experiment Settings

Following the tradition of the traffic signal control study [38], we conduct experiments in a simulation platform CityFlow[3], which is an open-source traffic simulator designed for large-scale traffic signal control [42]. The simulator takes traffic data as input, and a vehicle proceeds to its destination according to the simulation settings. The simulator can provide observations of the traffic situation to signal control methods and execute signal control actions. Similar to real world, a three-second yellow signal and a two-second all-red signal are set after each green signal.

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

In a multi-intersection network setting, we use the real road network to define the network in the simulator. For a single intersection, unless otherwise specified, the road network is set to be a four-way intersection, with four 300-meter long road segments.

The code will be accessed on the authors' website.

5.2 Datasets

We use two private real-world datasets from Jinan and Hangzhou in China and one public dataset from Atlanta in the United States.

Jinan. We collect data from our collaborators in Jinan from surveillance cameras near intersections. There are in total 7 intersections with relatively complete camera records for single intersection control. Each record in this dataset contains the camera location, the time when one vehicle arrived at the intersection, and the vehicle information. These records are recovered from the camera recordings by advanced computer techniques. We feed the vehicles to the intersections at their recorded arrival time in our experiments.

Hangzhou. This dataset is captured by surveillance cameras in Hangzhou from 04/01/2018 to 04/30/2018. There are in total 6 intersections with relatively complete camera records. These records are processed similarly as the Jinan data.

Atlanta. This public dataset[4] is collected by eight video cameras from an arterial segment on Peachtree Street in Atlanta, GA, on November 8, 2006. This vehicle trajectory dataset provides the precise location of each vehicle within the study area and five intersections in total are taken into consideration.

5.3 Methods for Comparison

To evaluate the effectiveness and efficiency of our model, we compare it with the following classic and state-of-the-art methods. We tune the parameters of each method separately and report the best performance obtained.

- **Fixedtime** [23]: Fixed-time control adopts a pre-determined cycle and phase time plan, which is widely used in the steady traffic flow. A grid search is conducted to find the best cycle.
- **SOTL** [11]: Self-Organizing Traffic Light Control is an approach which can adaptively regulate traffic lights based on a hand-tuned threshold on the number of waiting vehicles.

[3] https://github.com/cityflow-project/CityFlow
[4] https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

Table 1: Overall performance. Travel time is reported in the unit of second.

Model	Jinan							Hangzhou					
	1	2	3	4	5	6	7	1	2	3	4	5	6
Fixedtime	118.82	250.00	233.83	297.23	101.06	104.00	146.66	271.16	192.32	258.93	207.73	259.88	237.77
Formula	107.92	195.89	245.94	159.11	76.16	100.56	130.72	218.68	203.17	227.85	155.09	218.66	230.49
SOTL	97.80	149.29	172.99	64.67	76.53	92.14	109.35	179.90	134.92	172.33	119.70	188.40	171.77
DRL	98.90	235.78	182.31	73.79	66.40	76.88	119.22	146.50	118.90	218.41	80.13	120.88	147.80
IntelliLight	88.74	195.71	100.39	73.24	61.26	76.96	112.36	97.87	129.02	186.04	81.48	177.30	130.40
A2C	135.81	166.97	226.82	43.28	67.05	148.69	236.17	110.91	98.56	187.41	86.56	116.70	128.88
FRAP	**66.40**	**88.40**	**84.32**	**33.83**	**54.43**	**61.72**	**72.31**	**80.24**	**79.43**	**110.33**	**67.87**	**92.90**	**88.28**
Improvement	25.17%	40.79%	16.01%	47.69%	11.15%	19.72%	33.87%	18.01%	33.20%	35.98%	15.30%	23.15%	32.30%

Table 2: Overall performance.

Model	Atlanta				
	1	2	3	4	5
Fixedtime	140.51	334.17	334.01	353.56	271.14
Formula	116.16	148.71	163.93	157.08	254.00
SOTL	101.87	133.79	136.77	138.75	73.93
DRL	152.93	95.83	101.61	79.03	43.04
IntelliLight	76.25	74.10	83.12	65.94	47.51
A2C	147.01	105.16	87.42	70.99	49.23
FRAP	**67.45**	**61.48**	**65.26**	**60.75**	**41.39**
Improvement	11.54%	17.03%	21.49%	7.87%	3.83%

- **Formula**: This method computes a reasonable cycle length of the traffic signal from the traffic condition, i.e., the preset volume for a uniform flow. Then the time assigned to each phase is decided by the traffic volume ratio.
- **DRL** [32]: This method leverages a DQN framework for traffic light control and takes as state an image depicting vehicles' positions on the road.
- **IntelliLight** [38]: This is another deep reinforcement learning method with a more elaborate network architecture. This is the state-of-the-art RL method and demonstrates good performance in 2-phase signal control.
- **A2C** [10]: This method adopts the state-of-the-art advantage actor-critic framework and utilizes waiting time and the number of vehicles to depict traffic states.

5.4 Parameter Settings

To train our models, some hyperparameters are tuned for best performance. We adopt an Adam optimizer with learning rate 1e-3. For each training round, 1000 transitions are sampled from memory and trained with batch size 20. Three actors run in parallel to implement the Ape-X DQN framework. For the time interval between two consecutive actions, we carry out a simple experiment and find model performance is not sensitive to this parameter. As shown in Figure 7, travel time is not much affected by time interval. Therefore, we just set it to be 10s as suggested in a recent work [14].

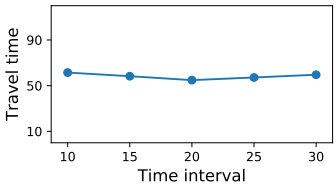


Figure 7: Effects of the time interval.

5.5 Evaluation Metrics

Based on existing studies in traffic signal control, we choose a representative metric, **travel time**, for evaluation. This metric is defined as the average travel time that vehicles spend on approaching lanes (in seconds), which is the most frequently used measure to judge performance in the transportation field.

5.6 Overall Performance

Table 1 and 2 report the travel times achieved by all methods with the 8-phase setting. Note that **Improvement** is the percentage by which FRAP surpasses the best baseline. We can see clearly that our method significantly outperforms all other methods on all datasets.

As expected, RL methods tend to perform better than conventional ones like Fixedtime as the ability to capture real-time information at the intersection enables RL methods to make more reasonable decisions. Among these RL approaches, our method stands out not only in terms of travel time, but also in terms of convergence speed. Figure 8 plots the convergence curves of RL methods and FRAP leads to the fastest convergence (we only show one case due to space limitations). It is because FRAP leverages the symmetry properties of traffic signal control and the Ape-X DQN framework to improve sampling efficiency.

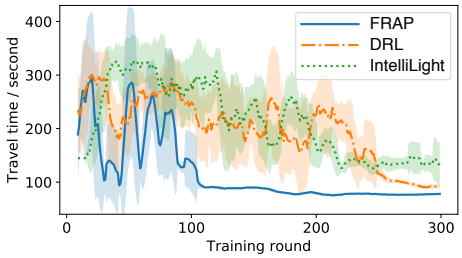


Figure 8: Convergence speed of RL methods.

5.7 Model Characteristics

5.7.1 Invariance to flipping & rotation. Besides achieving faster travel time and convergence speed, FRAP has another advantage in its invariance to flipping and rotation. In the real world, it is common that people drive to work in a specific movement in the morning and go home in the opposite direction in the afternoon. Figure 9 shows an example of traffic flow flipping from intersection 4 in Jinan. It can be observed that the traffic volume of the west approach is much larger than that of the east approach at around 8 am. At 5 pm, the relation is reversed.

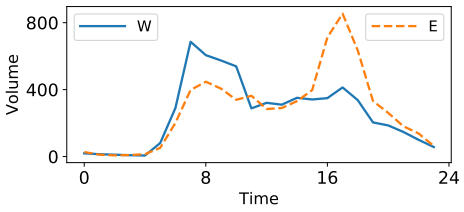
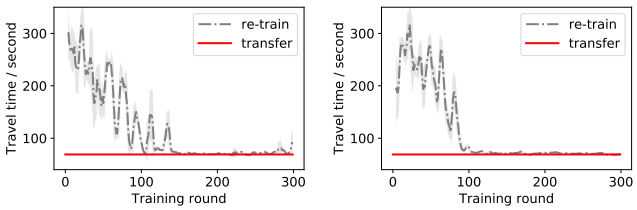


Figure 9: An instance of traffic flow flipping from the intersection 4 in Jinan, China.

Without using the symmetry properties of traffic signal control, previous RL methods have to re-train a model when the flow varies drastically (e.g., flipping and rotation). However, the FRAP model design guarantees that our method is less vulnerable to such extreme changes, meaning that the model will perform nearly the same under those traffic flows. Figure 10 shows FRAP's invariance to flipping and rotation. In this experiment, we directly take a model trained from the original traffic flow and test it on the flipped and rotated flows. We compare its performance with two re-trained models. From this figure, we can observe that our transferred model achieves almost identical travel time performance to the re-trained models, thus spares the extra training costs.



(a) Flipping (b) Rotation

Figure 10: Invariance to flipping and rotation. We show the convergence curves of re-trained and transferred models for flipped and rotated traffic flow.

5.7.2 Adaptation to different traffic volumes. In this experiment, we illustrate another advantage of FRAP over other RL methods in its adaptability to traffic volume. Intuitively, if a model explores sufficient states when trained on a heavy traffic flow, it can adapt to those relatively light flows. However, due to the large state space, existing RL methods can hardly see enough samples for the transfer to different traffic flows. In the meantime, the FRAP model design takes advantage of the symmetry properties of traffic signal control to improve data efficiency, which leads to better transferability.

In this experiment, we choose both FRAP and IntelliLight models trained from the intersection 2 in Jinan with the largest vehicle volume, and evaluate their performances on a relatively light flow from the intersection 1. In this case, both models have explored similar states and converged to the best values they can achieve for intersection 2. From Figure 11, we can see that the transferred model of FRAP performs almost the same as the re-trained model, whereas there is a distinct gap between re-trained and transferred models of IntelliLight. This suggests that the proposed FRAP model design increases sampling efficiency significantly and facilitates adaptation to different traffic flows.

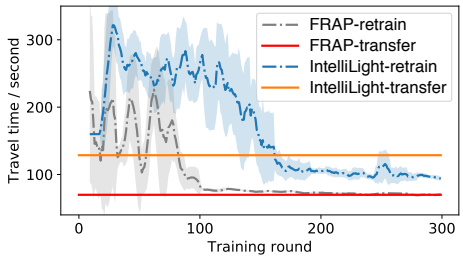


Figure 11: FRAP model design leads to higher data efficiency and better adaptation to different traffic volumes.

5.7.3 Flexibility of the 8-phase setting. Due to transportation traditions, an intersection with the 4-phase setting (Phase A, D, E, and H) is also very common in the real world. Although the 4-phase setting is simple and efficient sometimes, it has a severe limitation in that the green time for through or left-turn signal is always the same in the two opposite approaches. This will exert negative effects on travel time when the volume is unbalanced in these two approaches, which occurs frequently on real roads. Meanwhile, the 8-phase setting allows vehicles from one approach to pass exclusively. In this experiment, we show that the 8-phase setting is more flexible than the 4-phase setting, that is, under the 8-phase setting, vehicles can pass faster and more reasonably.

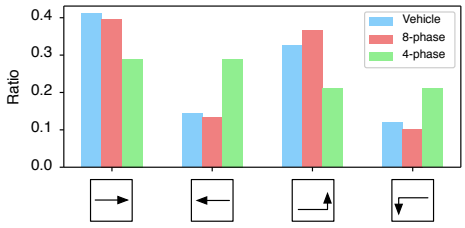


Figure 12: Traffic volume ratio and green time ratio (under the 4-phase and 8-phase settings) of W and E approach movements at the intersection 1 in Jinan, China.

As shown in Figure 12, when the vehicle volume is quite different on west and east approaches at the intersection 1 in Jinan, the policy learned under the 8-phase setting can adjust its green time accordingly, whereas the policy under 4-phase setting wastes a significant amount of time on the movement with light traffic(e.g., the east-through movement). Indeed, travel time is only 66.40s

under the 8-phase setting, but increases to 81.97s under the 4-phase setting. Thus, the general 8-phase setting brings flexibility to traffic signal control as it adapts better to unbalanced traffic flows.

5.8 Experiment on Different Environments

5.8.1 Experiment on different intersection structures. We also evaluate the performance of our method under different intersection structures (i.e., 3-, 4-, and 5-approach intersections). For this experiment, we synthesize one typical traffic flow for each structure based on the mean and variance of traffic flows in Jinan, China.

We modify our model as follows. For the 3-approach intersection, we disable some neurons in our network to make it compatible with this structure. In addition, the input is modified with zero padding for nonexistent movements. For the 5-approach structure, we add another phase to the model, and then use the same process (i.e., phase demand modeling, phase pair embedding, and phase pair competition) as in Figure 4 to predict the Q-value of all phases.

Detailed results of different intersection structures are reported in Table 3. We can see that FRAP performs consistently better than other methods and can be applied easily to all structures without major modification.

Table 3: Performance on different intersection structures.

Model	3-approach	4-approach	5-approach
Fixedtime	166.16	93.21	211.26
Formula	159.12	67.17	231.33
SOTL	123.43	65.39	124.71
DRL	108.94	125.84	140.33
IntelliLight	108.27	60.38	151.92
A2C	132.56	52.64	172.60
FRAP	**81.57**	**48.83**	**110.66**

5.8.2 Extension to a multi-intersection environment. Compared with a single intersection, people sometimes concern more about the overall traffic light control for an area containing multiple intersections. A straightforward way to enable intelligent traffic light control in the multi-intersection environment is to assign an independent RL agent for each intersection.

Table 4: Performance in a multi-intersection environment.

Model	Jinan	Hangzhou	Atlanta
Fixedtime	880.18	823.13	493.49
Formula	385.46	629.77	831.34
SOTL	1422.35	1315.98	721.15
DRL	1047.52	1683.05	769.46
IntelliLight	358.83	634.73	306.07
A2C	316.61	591.14	244.10
FRAP	**293.35**	**528.44**	**124.42**

To validate the potential of our model in the multi-intersection environment, we select a 3×4, a 4×4, and a 1×5 grid of intersections in Jinan, Hangzhou, and Atlanta respectively. The Jinan and Hangzhou data are selected from regions with relatively rich data coverage. Necessary missing data filling in are done to preprocess the data. The performance of different methods is shown in Table 4. We can see that FRAP stands out among all traffic signal control methods again in this setting. For further improvement, coordination of neighboring intersections can be considered as a promising direction for future work.

5.9 Interpretation of Learned Policies

To gain additional insight about what FRAP has learned, we choose one intersection and visualize the learned policy in the following way: for each hour between 8 am and 8 pm, the busiest time in a day, we calculate the green light time assigned to each movement according to the specific policy and then normalize it to obtain the green time percentage. In the meantime, we compute the ratio of vehicle volume on each movement for reference.

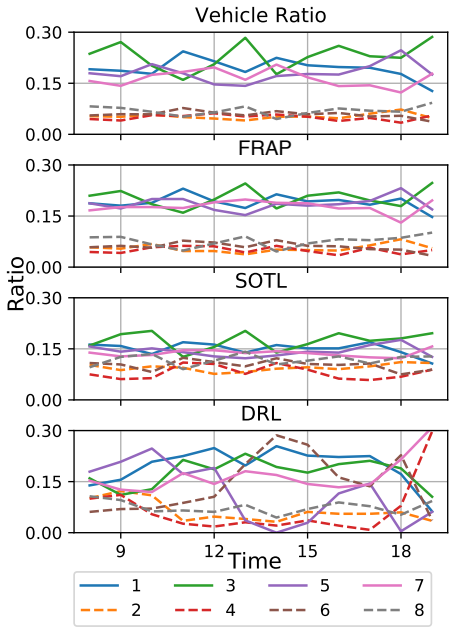


Figure 13: Traffic volume ratio and green time ratio of three traffic signal control methods for each movement between 8 am and 8 pm. The number from 1 to 8 indicates a specific movement signal described in Figure 3(b).

As shown in Figure 13, the green time ratio of FRAP synchronizes well with the percentage of traffic volume in each hour, while other baseline methods such as SOTL and DRL would allocate green light time to each movement more randomly and irregularly. Specifically, during the selected period, the vehicle volume of four left-turn movements is relatively light compared with that of four through ones. Thus, a good policy is expected to assign more green time to the through movements. Figure 13 shows that FRAP indeed divides the through and left-turn movements into two groups, whereas SOTL and DRL largely mix them together.

6 CONCLUSION

In this paper, inspired by the universal principles of competition and invariance, we propose a novel RL model FRAP for traffic

signal control. We analyze the advantage of FRAP over other RL methods in sampling efficiency and carry out comprehensive experiments on three datasets. Results demonstrate that our method converges faster and achieves better performance than state-of-the-art methods. Furthermore, we show the potential of our model in handling complex scenarios such as different intersection structures and multi-intersection environments. For future work, patterns of pedestrians and non-motorized vehicles need to be considered and a field study can be an important step for our model to get real-world feedback and for us to validate the proposed RL approach.

ACKNOWLEDGMENTS

The work was supported in part by NSF awards #1652525, #1618448 and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. 2011. Traffic light control in non-stationary environments based on multi agent Q-learning. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 1580–1585.

[2] Monireh Abdoos, Nasser Mozayani, and Ana LC Bazzan. 2014. Hierarchical control of traffic signals using Q-learning with tile coding. *Applied intelligence* 40, 2 (2014), 201–213.

[3] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering* 129, 3 (2003), 278–285.

[4] Mohammad Aslani, Mohammad Saadi Mesgari, Stefan Seipel, and Marco Wiering. 2018. Developing adaptive traffic signal control by actor–critic and direct exploration methods. In *Proceedings of the Institution of Civil Engineers-Transport*. Thomas Telford Ltd, 1–10.

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

[6] Bram Bakker, M Steingrover, Roelant Schouten, EHJ Nijhuis, LJHM Kester, et al. 2005. Cooperative multi-agent reinforcement learning of traffic lights. (2005).

[7] Tim Brys, Tong T Pham, and Matthew E Taylor. 2014. Distributed learning and multi-objectivity in traffic light control. *Connection Science* 26, 1 (2014), 65–83.

[8] Vinny Cahill et al. 2010. Soilse: A decentralized approach to optimization of fluctuating urban traffic using reinforcement learning. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 531–538.

[9] Noe Casas. 2017. Deep Deterministic Policy Gradient for Urban Traffic Light Control. *arXiv preprint arXiv:1703.09035* (2017).

[10] 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).

[11] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. 2013. Self-organizing traffic lights: A realistic simulation. In *Advances in applied self-organizing systems*. Springer, 45–55.

[12] Samah El-Tantawy and Baher Abdulhai. 2010. An agent-based learning towards decentralized and coordinated traffic signal control. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 665–670.

[13] Samah El-Tantawy and Baher Abdulhai. 2012. Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC). In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 319–326.

[14] 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.

[15] Martin Fellendorf. 1994. VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In *64th Institute of Transportation Engineers Annual Meeting*. Springer, 1–9.

[16] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. 2018. Distributed Prioritized Experience Replay. *CoRR* abs/1803.00933 (2018).

[17] 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).

[18] 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.

[19] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. 2018. Deep reinforcement learning for traffic light control in vehicular networks. *arXiv preprint arXiv:1803.11115* (2018).

[20] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

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

[22] Patrick Mannion, Jim Duggan, and Enda Howley. 2016. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic Road Transport Support Systems*. Springer, 47–66.

[23] Alan J. Miller. 1963. Settings for Fixed-Cycle Traffic Signals. *Journal of the Operational Research Society* 14, 4 (01 Dec 1963), 373–386.

[24] Pitu Mirchandani and Larry Head. 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies* 9, 6 (2001), 415–432.

[25] Pitu Mirchandani and Fei-Yue Wang. 2005. RHODES to intelligent transportation systems. *IEEE Intelligent Systems* 20, 1 (2005), 10–15.

[26] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. 1928–1937.

[27] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. 2017. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *Intelligent Transport Systems (ITS)* 11, 7 (2017), 417–423.

[28] Tong Thanh Pham, Tim Brys, Matthew E Taylor, Tim Brys, Madalina M Drugan, PA Bosman, Martine-De Cock, Cosmin Lazar, L Demarchi, David Steenhoff, et al. 2013. Learning coordinated traffic light control. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-13)*, Vol. 10. IEEE, 1196–1201.

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

[30] As' ad Salkham, Raymond Cunningham, Anurag Garg, and Vinny Cahill. 2008. A collaborative reinforcement learning approach to urban traffic control optimization. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*. IEEE Computer Society, 560–566.

[31] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.

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

[33] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[34] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. 2015. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581* (2015).

[35] FV Webster and BM Cobbe. 1966. Traffic Signals. *Road Research Technical Paper* 56 (1966).

[36] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. CoLight: Learning Network-level Cooperation for Traffic Signal Control. *arXiv preprint arXiv:1905.05717* (2019).

[37] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A Survey on Traffic Signal Control Methods. *arXiv preprint arXiv:1904.08117* (2019).

[38] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 2496–2505.

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

[40] Lun-Hui Xu, Xin-Hai Xia, and Qiang Luo. 2013. The study of reinforcement learning for traffic self-adaptive control under multiagent markov game environment. *Mathematical Problems in Engineering* 2013 (2013).

[41] Nan Xu, Guanjie Zheng, Kai Xu, Yanmin Zhu, and Zhenhui Li. 2019. Targeted Knowledge Transfer for Learning Traffic Signal Plans. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 175–187.

[42] 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. In *The World Wide Web Conference*. ACM, 3620–3624.

[43] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Diagnosing Reinforcement Learning for Traffic Signal Control. *arXiv preprint arXiv:1905.04716* (2019).