MonitorLight: Reinforcement Learning-based Traffic Signal Control Using Mixed Pressure Monitoring

Zekuan Fang
MoE Engineering Research Center of SW/HW Co-design Tech. and App., East China Normal University
Shanghai, China
51205902008@stu.ecnu.edu.cn

Fan Zhang
MoE Engineering Research Center of SW/HW Co-design Tech. and App., East China Normal University
Shanghai, China
51205902104@stu.ecnu.edu.cn

Ting Wang
MoE Engineering Research Center of SW/HW Co-design Tech. and App., East China Normal University
Shanghai, China
twang@sei.ecnu.edu.cn

Xiang Lian
Department of Computer Science, Kent State University
Kent, USA
xlian@kent.edu

Mingsong Chen
MoE Engineering Research Center of SW/HW Co-design Tech. and App., East China Normal University
Shanghai, China
mschen@sei.ecnu.edu.cn

ABSTRACT

Although Reinforcement Learning (RL) has achieved significant success in the Traffic Signal Control (TSC), most of them focus on the design of RL elements while the impact of the phase duration is neglected. Due to the lack of exploring dynamic phase duration, the overall performance and convergence rate of RL-based TSC approaches cannot be guaranteed, which may result in poor adaptability of RL methods to different traffic conditions. To address these issues, in this paper, we formulate a novel *phase-duration-aware TSC* (PDA-TSC) problem and propose an effective RL-based TSC approach, named MonitorLight. Our approach adopts a new traffic indicator, *mixed pressure*, which enables RL agents to simultaneously analyze the impacts of stationary and moving vehicles on intersections. Based on the observed mixed pressure of intersections, RL agents can autonomously determine whether or not to change the current signals in real-time. In addition, MonitorLight can adjust the control method for scenarios with different real-time requirements and achieve excellent results in different situations. Extensive experiments on both real-world and synthetic datasets demonstrate that MonitorLight outperforms the current state-of-the-art IPDALight by up to 2.84% and 5.71% in average vehicle travel time, respectively. Moreover, our method significantly speeds up the convergence, leading IPDALight by 36.87% and 34.58% in the start to converge episode and jumpstart performance, respectively.

CCS CONCEPTS

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

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557400

KEYWORDS

Reinforcement learning; Traffic signal control; Phase duration; Average travel time; Fairness

ACM Reference Format:
Zekuan Fang, Fan Zhang, Ting Wang, Xiang Lian, and Mingsong Chen. 2022. MonitorLight: Reinforcement Learning-based Traffic Signal Control Using Mixed Pressure Monitoring. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557400

1 INTRODUCTION

Due to rapid urban growth and fast increase in vehicle numbers, people in large metropolitan areas are experiencing more and more serious traffic congestion problems. According to a recent survey [1] by INRIX Inc. (a world leader in transportation analytics) in 2021, the per capita traffic congestion loss is more than 130 hours for top-3 world's most congested cities. In other words, commuters in these cities lost more than five days on average per year due to traffic congestion issues. Therefore, how to alleviate the impact of traffic congestion has become a top public policy challenge for urban governance. As one of the most effective methods to govern the traffic congestion, Traffic Signal Control (TSC) has been extensively studied [2]. Reasonable TSC strategies can maximize the traffic flow and effectively alleviate the road congestion. Traditional TSC methods such as Fixedtime [3] can handle most scenarios where the traffic flow is stable. However, with the huge increase in the number of vehicles, the traffic situation has become increasingly complex and dynamic, which makes it difficult to use the fixed signal change sequence and fixed signal duration method.

Recently, with the upgrading of Road Side Units (RSU) [4] and sustainable development of Artificial Intelligence (AI) technology, Reinforcement Learning (RL) has been studied largely in the design of control components for traffic-oriented Cyber-Physical Systems (CPS) [5], especially for TSC [6–9]. Similar to RL methods in other fields, the environment interaction and trial-and-error learning methods enable the RL-based TSC method to adapt to changes in the traffic flow and continuously improve its strategy. This makes

RL's control effect in dynamic traffic flow scenarios significantly better than traditional methods. In addition, most RL methods adopt the experience replay [10] to improve the performance.

The action in RL methods consists of the selection of signals and signal durations. However, most RL-based TSC methods only consider selecting better signals with appropriate state and reward design, while ignoring the selection of the signal durations. Note that, the fixed signal duration can cause the same state and phase to transfer to different new states, and may reduce the sample efficiency which plays a crucial role in the learning efficiency for most machine learning processes. The lack of the signal duration design makes many RL methods with the fixed duration inefficient to learn and even fail to converge, which has a negative impact on RL methods that require dynamic adaptation to traffic flow changes.

To explore the impact of the phase duration on TSC and improve the learning efficiency and control effect of RL, this paper formulates a novel phase-duration-aware TSC problem (PDA-TSC problem) and presents an RL-based TSC approach, named MonitorLight, which monitors and analyzes the phase effect on intersections in real-time. We make the following contributions.

- We analyze the impact of the phase duration design on RL-based TSC methods, and formulate a novel PDA-TSC problem. To solve this problem, we propose an effective RL-based TSC method, MonitorLight, which focuses on controlling phase duration and makes the agent determine whether the phase needs to be changed in real-time based on our monitoring attribute.
- We present the reward design and state representation based on our proposed concept of mixed pressure, which comprehensively considers the influence of vehicle dynamics and vehicle waiting time, helps analyze the intersection state in the current phase, and enables the agent to better model the traffic situation.
- We evaluate our proposed MonitorLight on both real-world and synthetic traffic datasets. Through extensive experiments, we demonstrate that our approach outperforms the state-of-the-art methods in the TSC performance and convergence rate.

2 RELATED WORK

The conventional TSC methods are often heuristic algorithms designed based on historical experience [11]. These methods generally set a fixed total period and signal change sequence, optimizing the green signal duration of each phase and the phase offset between intersections based on classic algorithms in the transportation field. For example, Cools et al. [12] proposed a method called SOTL, which changes traffic signals based on the waiting vehicle count. Koonce and Rodegerdts [13] proposed Webster, which calculates the optimal cycle length and phase duration by analyzing factors such as traffic efficiency, capacity, and phase types of intersections. However, the dependence on uniform traffic flow makes these methods inappropriate in the complex and dynamic traffic environment.

To dynamically adjust the control strategy with dynamic traffic flow, various RL-based TSC methods have been proposed [14]. Typically, RL methods treat the intersection situation as RL state and the signal control as RL action. Compared with conventional methods, RL methods do not need to collect a large amount of historical data for computational analysis, while relying on environmental interaction and autonomous learning capabilities to achieve better results in complex traffic flow conditions. For example, Wei et al. [15] combined Graph Neural Network (GNN) [16] and RL to model the traffic network so that the signal selected by the agent can consider the impact of adjacent intersections on the target intersection, improving the coordination between intersections. To improve the generalization ability of RL models, Liu et al. [17] designed a traffic flow generator based on Wasserstein generative adversarial network to improve the adaptability of RL models to dynamic traffic flows. Ye et al. [18] considered the collaborative optimization between intersections and proposed FedLight, which combines federated learning and RL. Jiang et al. [19] searched for optimal TSC and dynamic lane control solutions based on group attention and multi-timescale learning. Yu et al. [20] proposed the MaCAR framework, which uses Message Propagation Neural Network (MPNN) [21] and Traffic Forecasting Network (TFN) to improve the communication between agents. Inspired by Max Pressure (MP) [22], Wei et al. [23] proposed an RL method based on the intersection pressure named PressLight. The theory-based state and reward design significantly reduce the average vehicle travel time. However, these methods only consider the appropriate design of the state representation, while ignoring the design of the phase duration and adopting a fixed phase duration. Fixed phase duration will affect the flexibility of the agent's strategies, making it difficult for each action to achieve the best effect, which in turn affects the TSC control effect.

To further improve the flexibility of RL strategies, various dynamic duration RL methods have been proposed. For example, Hu et al. [24] considered the changes in traffic flow at intersections over time. Each agent combines the GNN and Convolutional Neural Network (CNN) to predict the time required for all vehicles to pass through the intersection and sets this time as phase duration. Bouktif et al. [25] used P-DQN [26] and regarded the TSC system as a hybrid system. This RL strategy divided the action into discrete action and continuous action, where discrete action corresponded to the selected phase, and continuous action corresponded to the phase duration. Zhao et al. [27] proposed the concept of Intensity by vehicle dynamics and used this concept to model the intersection and dynamically adjust the phase duration based on a heuristic. Although the dynamic duration improves the flexibility of the RL strategies, it still cannot guarantee that each phase can achieve the best phase effect. This is because the traffic flow is complex and dynamic, and the agent cannot foresee when the phase will reach the best phase effect.

To the best of our knowledge, our work is the first attempt to formulate the impact of phase duration on the RL training process and implement RL methods based on real-time phase effect monitoring, which makes each action can achieve the best effect and improve the TSC performance and convergence rate.

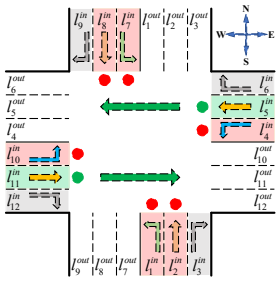
3 PROBLEM DEFINITION

This section introduces the preliminaries about TSC, and formally defines the phase-duration-aware TSC problem.

3.1 Preliminaries

This subsection will explain some key concepts in TSC, which are widely accepted by previous work. Figure 1 shows a classic TSC environment. The left part is an intersection, where green lanes

indicate action lanes, red lanes indicate non-action lanes, and grey lanes indicate the right-turning lanes. The right part is a set of four typical phases and their corresponding action and non-action lanes.



Phase	Movement	Action Lane	Non-action Lane
SN-straight		l_2, l_8	$l_1, l_4, l_5, l_7, l_{10}, l_{11}$
WE-straight		l_5, l_{11}	$l_1, l_2, l_4, l_7, l_8, l_{10}$
SN-left		l_1, l_7	$l_2, l_4, l_5, l_8, l_{10}, l_{11}$
WE-left		l_4, l_{10}	$l_1, l_2, l_5, l_7, l_8, l_{11}$

Figure 1: An illustration of the intersection environment.

Definition 3.1. ***(Road Network)*** *A road network is an environment in which vehicles travel, consisting of multiple intersections and several roads connecting the intersections.*

Definition 3.2. ***(Intersection)*** *An intersection contains roads in multiple directions, and each road contains a number of lanes.*

As shown in Figure 1, the intersection includes roads in four directions, and each road consists of three lanes, which are left-turning lanes, straight-going lanes, and right-turning lanes.

Definition 3.3. ***(Incoming and Outgoing Lanes)*** *An incoming lane is a lane where vehicles enter an intersection. An outgoing lane is a lane where vehicles exit an intersection. The set of incoming lanes and outgoing lanes are denoted as L^{in} and L^{out} respectively.*

As depicted in the left part of Figure 1, L^{in} contains 12 incoming lanes ($\{l_1^{in}, l_2^{in}, ..., l_{12}^{in}\}$) and L^{out} contains 12 outgoing lanes ($\{l_1^{out}, l_2^{out}, ..., l_{12}^{out}\}$) in this intersection.

Definition 3.4. ***(Traffic Movement)*** *For each intersection, traffic movement is vehicles passing through the intersection from an incoming lane to an outgoing lane.*

Definition 3.5. ***(Signal and Phase)*** *Signal determines whether a traffic movement is allowed at the current time. The red signal indicates the prohibition, and the green signal allows moving. A phase is a combination of signals at the current time, and all signals in a phase do not conflict with each other.*

As shown on the left of Figure 1, green straight arrows indicate the currently permitted traffic movements, and red and green dots indicate signals. Note that, vehicles in the right-turning lanes can turn right at any time, so it is meaningless to set signals for right-turning lanes. As shown on the right of Figure 1, there are four phases at an intersection, including South-North going straight (SN-straight), West-East going straight (WE-straight), South-North turning left (SN-left), and West-East turning left (WE-left).

Definition 3.6. ***(Action and Non-Action Lanes)*** *For the current phase a, an action lane is an incoming lane that allows vehicles to move (except for right-turning lanes), which is denoted as l_{act}. A non-action lane is an incoming lane that forbids vehicles to move, which is denoted as l_{nac}. The set of action lanes is denoted as L_{act}^a, and the set of non-action lanes is denoted as L_{nac}^a.*

Consider an example of the WE-straight intersection phase in Figure 1, L_{act}^a contains $\{l_5^{in}, l_{11}^{in}\}$ and L_{nac}^a contains $\{l_1^{in}, l_2^{in}, l_4^{in}, l_7^{in}, l_8^{in}, l_{10}^{in}\}$.

Definition 3.7. ***(Stationary and Moving Vehicles)*** *For the vehicles set V_i in lane l_i, vehicles with speeds equal to 0 m/s are stationary vehicles V_i^s, and vehicles with speeds more than 0 m/s are moving vehicles V_i^d.*

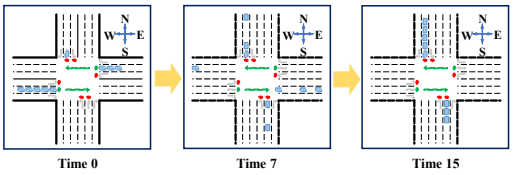
Definition 3.8. ***(Intersection Pressure)*** *The intersection pressure refers to the difference between the numbers of vehicles on all outgoing and incoming lanes at the intersection.*

Problem Setting. This paper focuses on the performance and convergence speed of RL methods on different traffic flow datasets. Our approach improves the learning efficiency of RL by solving the problem of phase duration design. In addition, our approach can also address the fairness problem (explained in Section 5.7), which is a common problem in many RL methods.

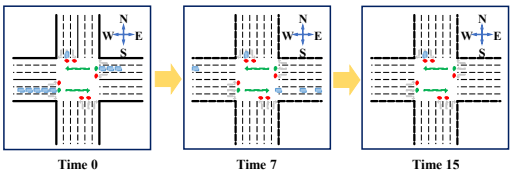
3.2 The PDA-TSC Problem

This subsection will analyze the impact of unreasonable phase duration design on RL-based TSC methods, and formulate a *phase-duration-aware traffic signal control* (PDA-TSC) problem.

The impact of phase duration on the RL training process. The sample efficiency has an important impact on the learning efficiency of RL. In the TSC field, samples are generated by RL agents interacting with the environment. Most of the RL-based TSC methods usually use a fixed phase duration. When switching phases, the agent selects action A (phase) according to the current intersection state S, and after a fixed period of time, gets a reward R and a new state S'. The RL agent stores $\langle S, A, R, S' \rangle$ in memory and update RL model by training.



(a) A WE-straight phase with vehicles entering the intersection.



(b) A WE-straight phase with no vehicle entering the intersection.

Figure 2: Example of RL methods with fixed phase duration.

However, the reward R after a fixed phase duration is not necessarily the optimal reward that action A (phase) can achieve. Taking Figure 2 as an example, the RL agents in Figures 2(a) and Figure 2(b) select the same phase (WE-straight) because of the same state S_0 and model parameters θ. Assume that the fixed phase duration here is 15s. Within this duration, vehicles enter the intersection continuously in the south-north direction in Figure 2(a), resulting in a low reward R_a of environmental gives back at the end of the phase. In contrast, in Figure 2(b), no vehicle enters the intersection

in the south-north direction, so at the end of the phase, the environment gives back a high reward R_b. This results in two different transitions under the same state S_0 and action A_0.

$$Transition_a = \langle S_0, A_0, R_a, S'_a \rangle, \tag{1}$$

$$Transition_b = \langle S_0, A_0, R_b, S'_b \rangle. \tag{2}$$

If these two transitions are used for model training simultaneously, it will produce a situation where the input (S_0, A_0) is the same, but the difference between target values Y_a and Y_b is rather large,

$$Y_a = R_a + \gamma \cdot argmax_{act} Q(S'_a, act; \theta), \tag{3}$$

$$Y_b = R_b + \gamma \cdot argmax_{act} Q(S'_b, act; \theta), \tag{4}$$

which leads to the large difference between $Loss_a$ and $Loss_b$.

$$Loss_a = [Y_a - Q(S_0, A_0; \theta)]^2, \tag{5}$$

$$Loss_b = [Y_b - Q(S_0, A_0; \theta)]^2. \tag{6}$$

This will update the model parameters to different degrees or even in opposite directions, which will make it difficult for the model to quickly approach the optimal solution, and even cause the model to fail to converge.

Formulation of the phase-duration-aware TSC problem. Inspired by the example above, we can find that it is essential to select an appropriate duration for the phase selection. Existing dynamic duration RL methods [24, 25, 27] analyze the number of vehicles on the action lane to calculate the phase duration. However, such methods still cannot avoid the situation as shown in Figure 2. An efficient way to solve this issue is to end each phase in time when the phase achieves the best effect. This can ensure that the reward corresponding to the phase must be an optimal value within the duration, thereby avoiding the situation where the same state and phase produce a large difference in rewards.

Therefore, in this paper, we propose and formulate the PDA-TSC problem, defined as follows.

Definition 3.9. ***(Phase-Duration-Aware Traffic Signal Control Problem, PDA-TSC Problem)*** *During the RL training process, given traffic conditions at intersections on road networks, a phase-duration-aware traffic signal control (PDA-TSC) problem determines the phase duration of TSC, such that the generated transitions can maximize the learning efficiency and control performance of RL, where the learning efficiency is measured by the convergence rate, and the control performance is measured by the average vehicle travel time.*

Definition 3.9 defines the PDA-TSC problem which adaptively adjusts the phase duration so that the learning efficiency and the control performance can be maximized. However, it is a grand challenge to evaluate whether a phase reaches the best effect. To our best knowledge, prior RL-based TSC methods did not use any indicator to evaluate the phase effect. Thus, we will propose a novel RL method, named MonitorLight, aiming to monitor and achieve the best real-time phase effect, as discussed later in Section 4.

4 OUR MONITORLIGHT APPROACH

This section will introduce our proposed MonitorLight approach for the PDA-TSC problem. First, we define a new state representation that improves the environment modeling capabilities. Next, we introduce the design of the state, the action, and the reward. Finally, we explain the training process and testing process of MonitorLight. According to different real-time conditions, MonitorLight can be divided into MonitorLight-r and MonitorLight-p.

4.1 Mixed Pressure

This subsection will explain a new intersection state representation, mixed pressure, which comprehensively considers the impact of vehicle dynamics, vehicle waiting time, etc. on traffic efficiency.

PressLight [23] has proved that the traffic efficiency is related to the intersection pressure. Fewer stationary vehicles will lead to a shorter average vehicle travel time. Inspired by PressLight, we define static pressure which takes into account both the number of stationary vehicles and the waiting time of vehicles.

Definition 4.1. ***(Static Pressure)*** *For lane l_i, static pressure, P_s, is formulated as the sum of vehicle count and waiting time.*

$$P_s = \Sigma_{veh \in V_i^s} (1 + \omega \times t_s), \tag{7}$$

where V_i^s is the set of stationary vehicles on l_i, ω is an adjustable time coefficient which affects vehicle fairness (vehicle fairness is explained in Section 5.7), and t_s is the waiting time of veh before the intersection.

However, it is difficult to use a single attribute to evaluate the impact of vehicles on traffic efficiency. In addition to stationary vehicles, moving vehicles can also have an impact on intersection congestion. However, the affect of moving vehicles on intersection congestion depends on speeds and distances, so it is difficult to use a fixed value to evaluate it. In fluid mechanics, for the pressure of the fluid section, the higher flow rate and the lower density will lead to the lower pressure. Inspired by this, we treat the moving vehicle as a fluid and explore the effect of vehicle speed and distance on the lane pressure. We have carried out comprehensive experiments and the experimental results show that dynamic pressure is inversely proportional to the speed and the distance to the intersection.

Definition 4.2. ***(Dynamic Pressure)*** *For lane l_i, dynamic pressure, P_d, measures the effect of all moving vehicles at the intersection, which is formulated as:*

$$P_d = \Sigma_{veh \in V_i^d} \frac{1}{v \times \frac{L}{L_{max}} + 1}, \tag{8}$$

where V_i^d refers to the set of moving vehicles on l_i, v refers to the vehicle speed, L refers to the distance between the vehicle and the intersection, and L_{max} refers to the distance parameter which means the maximum distance that can be observed at the intersection. In addition, the 1 in the denominator ensures that the dynamic pressure generated by the moving vehicle must be less than the static pressure generated by the stationary vehicle, and when the vehicle speed is close to 0, the dynamic pressure is close to the static pressure.

We use the mixed pressure to represent the impact of stationary and moving vehicles on intersections.

Definition 4.3. ***(Mixed Pressure)*** *For lane l_i, mixed pressure, P_m, is formulated as the sum of static and dynamic pressure:*

$$P_m = P_s + P_d. \tag{9}$$

4.2 Agent Design

In this subsection, we introduce the design of RL agent, including the design of the state, the action, and the reward.

- **State**: $Agent_j$ observes part of the system environment as the state S_j. The state consists of the mixed pressure of all lanes. Take Figure 1 as an example, a four-way intersection has 12 incoming lanes and 12 outgoing lanes. In each step, $Agent_j$ observes the intersection, and calculates the mixed pressure of each lane. Therefore, we encode S_j as $(P^{l^i_1}_m, ..., P^{l^i_{12}}_m, -P^{l^o_1}_m, ..., -P^{l^o_{12}}_m)$, where $l^i \in L^{in}, l^o \in L^{out}$. Note that the state size can be adjusted according to the change of the lane count.
- **Action**: In TSC, the traffic light agent needs to select appropriate signals to maximize the traffic efficiency. According to Section 3.1, each agent can choose one of the four actions, corresponding to the four phases, respectively. Note that, vehicles turn right without the phase restriction.
- **Reward**: The reward R_j of $Agent_j$ is the evaluation of the current action. It has been proved in Presslight [23] that minimizing the pressure at the intersection is equivalent to reducing the average vehicle travel time. Therefore, according to subsection 4.1, when the action switches, the reward for the $Agent_j$ is the negative number of the mixed pressure value on incoming lanes, which is denoted as $R_j = -\Sigma_{l^i \in L^{in}} P^{l^i}_m$.

4.3 Implementation of MonitorLight

This subsection proposes a new monitoring attribute for the phase duration aware, and details MonitorLight, a novel mixed pressure based RL method.

4.3.1 Monitoring attribute. The monitoring attribute indicates the effect of the phase at the current moment, which is an indicator to determine whether to switch the phase.

Definition 4.4. ***(Monitoring Attribute)*** *At time t_j, the monitoring attribute is the ratio of dynamic pressure on action lanes to the static pressure on non-action lanes.*

$$Monitoring_j = \frac{\Sigma_{l_i \in l_{act}} P_d}{\max_{l_i \in l_{nac}} P_s + \epsilon}, \tag{10}$$

where P_d and P_s represent dynamic and static pressures of the lane, respectively, and ϵ is a small positive constant.

The agent analyzes in real-time whether the monitoring attribute is less than the threshold, and ends the phase when monitoring attribute is less than the threshold and reselects a new phase.

4.3.2 Training process. Figure 3(a) shows the training framework of MonitorLight, where each agent contains a phase selection model and a duration prediction model. During training, the phase duration is not determined by the duration prediction model, but by the monitoring attribute. As shown in Figure 3(a), the agent selects a new phase at time t_0. When the phase lasts for some time (base duration), the agent starts to obtain the monitoring attribute in real-time. At time t_j, the monitoring attribute is less than the threshold α, so the agent immediately ends the current phase, and get the phase duration $t_r = t_j - t_0$. In the process of interacting with the environment, the agent uses the transition $\langle S, A, R, S' \rangle$ as the training sample of the phase selection model, and uses t_r as the label of the duration prediction model, so as to achieve the synchronization optimization of the phase selection model and the duration prediction model.

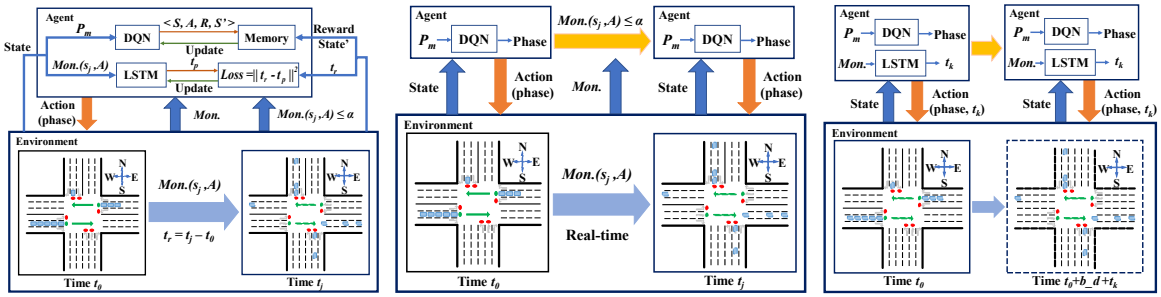
For the phase selection model, this paper uses the Deep Q Network (DQN). DQN combines deep learning and RL and uses neural networks to replace action value functions, which is essential to solve high-dimensional RL problems. In addition, this paper uses experience replay, which can eliminate the correlation of data. Experience replay reduces the variance of parameter updates, speeds up convergence, and improves the stability of RL training.

For the duration prediction model, this paper uses the classic time series model LSTM. The agent predicts the phase duration by obtaining a sequence of monitoring attributes for intersections within the base duration. And when the phase is switched, the loss is calculated with the real duration of the current phase to update the duration prediction model.

Algorithm 1: RL Implementation of MonitorLight

Input: i) E, # of episodes; ii) T, # of episode steps; iii) D, experience bank; iv) β, batchsize; v) γ, discount factor; vi) η, learning rate; vii) α, threshold; viii) ε, greedy coefficient; ix) b_d, base duration;

Initialize Q network with random parameters θ
$D \leftarrow \{\}$
for $i \leftarrow$ *1 to* E **do**
  ResetEnv()
  for $j \leftarrow$ *1 to* T **do**
    $s_j \leftarrow$ Observation()
    if $b\_d = 0$ *and* $Monitoring(s_j, A_k) \leq \alpha$ **then**
      $S' \leftarrow s_j$
      $R \leftarrow$ Calculate(s_j)
      $D \leftarrow D \cup \langle S, A_k, R, S' \rangle$
      $T \leftarrow j - k$
      if $Size(D) > \beta$ **then**
        $L \leftarrow \{\}$
        $B \leftarrow$ GetBatch(D)
        for $sample_k = \langle S_k, A_k, R_k, S'_k \rangle \in B$ **do**
          $Y_k \leftarrow R_k + \gamma * argmax_a Q(S'_k, a; \theta)$
          $\nabla_k \leftarrow [Y_k - Q(S_k, A_k; \theta)]^2$
          $L \leftarrow L \cup \nabla_k$
        end
        $Update\_Q(L)$
      end
      $Loss_t \leftarrow [T - LSTM(Monitoring(S, A_k))]^2$
      $Update\_LSTM(Loss_t)$
      $S \leftarrow s_j$
      $k \leftarrow j$
      if $Random(0,1) \leq \varepsilon$ **then**
        $A_k \leftarrow$ RandomAction()
      else
        $A_k \leftarrow argmax_a Q(S, a; \theta)$
      end
      ResetBaseDur()
    end
  end
end



(a) Training framework (b) MonitorLight-r (c) MonitorLight-p

Figure 3: An illustration of the proposed MonitorLight. Figure (a) is the training framework of MonitorLight, Figure (b) is the test framework of MonitorLight-r, and Figure (c) is the test framework of MonitorLight-p.

Algorithm 1 presents the training process of MonitorLight. Lines 1-2 initialize the Q network and flush the experience bank before learning. Lines 3-34 give the details of each episode. Line 4 resets simulation environment. Lines 5-33 show what the agent does at each step, where line 6 observes part of the environment and gets the intersection state. Line 7 gets the remaining base time and calculates the monitoring attribute. When the monitoring attribute is less than the threshold, the agent updates the Q network and reselects the action. In steps 8-10, the agent calculates the reward, and stores $\langle S, A, R, S' \rangle$ in the experience bank. Line 11, the agent gets the actual phase duration. Lines 12-21 represent the Q network update process. When the experience bank stores enough experience, line 15 randomly samples part of the experience. Lines 16-18 calculate the gradients based on these samples. The agent averages these gradients and updates parameters in line 20. Lines 22-23 update LSTM model. Steps 26-30 reselect the action (phase). Note that, the agent chooses the action based on the greedy coefficient to strengthen the agent's exploration ability.

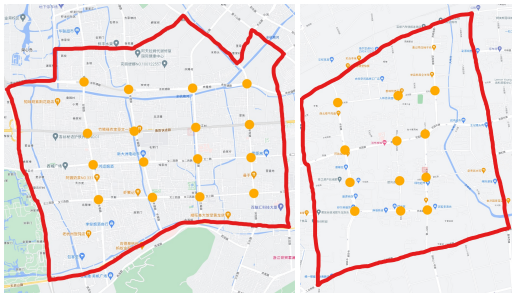
4.3.3 Testing process. According to different real-time conditions during testing, MonitorLight can be divided into MonitorLight-r based on real-time monitoring and MonitorLight-p based on phase duration prediction, and can switch between the two at any time.

- **MonitorLight-r**: As shown in Fig 3(b), phase switching is based on the real-time monitoring attribute. As with the training process, the current phase ends when the monitoring attribute is less than the threshold α. Then, the agent selects a new phase based on the mixed pressure at the intersection.
- **MonitorLight-p**: Without real-time monitoring, the agent needs to set the phase duration in advance. As shown in Fig 3(c), when the base duration ends, the monitoring attribute sequence is fed to the LSTM to get the phase extension time, i.e., $Duration = base\_duration + LSTM([Mon_0, ..., Mon_{b\_d}])$.

5 EXPERIMENTS

5.1 Environment Settings

To evaluate the effectiveness of our approach, we implemented MonitorLight on a Linux workstation with 3.70GHz Intel i9 CPU and 32GB memory. We used CityFlow [28], which provides an RL environment for large-scale city traffic scenarios, as the traffic simulator to simulate the road network. After inputting the traffic flow data, it will automatically generate vehicles driving along the designated path to the destination. In addition, the simulator can receive control information to change the phase at the intersection.



(a) Gudang, Hangzhou, China (b) Dongfeng, Jinan, China

Figure 4: Road networks for real-world datasets.

5.2 Datasets

We collected six datasets for the experiments as follows:

- **Synthetic Datasets:** We used four different sizes of synthetic datasets (Syn_1×3, Syn_2×2, Syn_3×3, Syn_4×4) in the experiment. Each intersection has four incoming roads and four outgoing roads, and each road has three lanes (turn left, turn right, go straight) with a lane length of 300 meters. The vehicle count in synthetic datasets conforms to the Gaussian distribution (with the mean 500 vehicles/hour/lane). The turning ratios are set as 10% (turn left), 60% (go straight), and 30% (turn right).
- **Real-World Datasets:** Two real-world datasets are actual traffic data in two cities (i.e., Hangzhou and Jinan). Their road networks are imported from GoogleMap, as shown in Figure 4. The (Red) polygons represent the selected areas, and the (yellow) dots indicate the intersections where traffic signals can be controlled. The sizes of Hangzhou and Jinan datasets are 4×4 and 4×3, respectively. The traffic flow data is collected from roadside surveillance cameras, and data statistics are depicted in Table 1.

Table 1: Data statistics of real-world traffic datasets

Dataset	Inter.#	Arrival rate (vehicles/300s)			
		Mean	Std	Max	Min
Hangzhou	16	526.63	86.70	676	256
Jinan	12	250.70	38.21	335	208

Table 2: Average travel time comparison on different datasets.

Type		Method	Synthetic Datasets				Real-World Datasets	
			Syn_1×3	Syn_2×2	Syn_3×3	Syn_4×4	Jinan	Hangzhou
Traditional		Fixed-Time	384.47	454.01	508.87	565.99	405.91	488.51
		SOTL	247.07	331.64	424.66	474.32	410.65	505.53
RL	Fixed	GRL	208.21	239.13	431.43	523.01	562.91	598.17
		CoLight	210.01	312.29	328.70	397.07	327.65	337.45
		PressLight	98.74	123.90	166.28	215.32	285.65	341.99
	Dynamic	IPDALight	88.01	109.66	146.92	184.54	255.35	298.99
Our		MonitorLight-r	**84.03**	**103.56**	**136.90**	**173.56**	**246.21**	**292.71**
		MonitorLight-p	87.64	109.16	143.44	183.52	251.33	296.79
Improvement over IPDALight			**4.52%**	**5.56%**	**6.82%**	**5.95%**	**3.58%**	**2.10%**

5.3 Baseline Methods

We compared our approach with two non-RL methods and four state-of-the-art RL methods:

- *Fixedtime* [11]: a classic method widely used in steady traffic flow scenarios, which cyclically changes the phase in a fixed order with a predefined duration.
- *SOTL* [12]: a classic method, which adaptively controls the traffic signal by checking whether the number of waiting vehicles on the road exceeds the threshold.
- *GRL* [7]: a fixed duration RL method for coordinated control of multiple intersections. By designing a coordination graph, the joint Q function of adjacent intersections is optimized, so as to achieve multi-intersection cooperation.
- *CoLight* [15]: a fixed duration RL method which is designed for the multi-intersection scenario. It combines deep RL with Graph Attention Network (GAT), which calculates the impact of the surrounding intersections to achieve coordinated control.
- *PressLight* [23]: a fixed duration RL method, with excellent results in both single- and multi-intersection scenarios. It designs RL elements based on the Max Pressure theory [22] to minimize the intersection pressure.
- *IPDALight* [27]: a dynamic duration RL method, which can effectively select control phases based on a new concept of Intensity. In addition, a heuristic algorithm is used to determine the phase duration while selecting the phase. The phase duration is equal to the ratio of the vehicles passing at the intersection to the total number of vehicles on incoming lanes.

We set the maximum number of training episodes for all RL methods to 200. In each episode, the batch size β is 32. The discount factor γ is 0.95. The learning rate η is 0.001. The greedy coefficient ε is 0.1. For fixed-duration RL methods, we set the phase duration to 15s. For IPDALight, the agent selects the duration within [5s, 25s] based on the calculated states of vehicles. For our approach, the time coefficient ω is 0 in the performance comparison, and the threshold α is 0.7. For the sake of fairness, we set the base duration of MonitorLight to 5s, which means the agent judges the monitoring attribute after the action lasts for 5s.

5.4 Evaluation Metrics

We evaluated the performance of TSC methods using the most commonly-used metric of **average travel time**, which refers to the average time of all vehicles staring from they enter the traffic network till they leave the traffic network. In addition, to compare the convergence performance of RL methods, we consider the metrics of **Start Episode number of Convergence (SEC)** [18] and **Jumpstart Performance (JP)** [15] for each RL method, where SEC represents the number of episodes when the RL model starts to converge and JP indicates the performance after the first episode.

5.5 Performance Comparison

Table 2 compares the average travel time between MonitorLight and state-of-the-art TSC methods. From this table, we can find that:

- RL-based methods outperform traditional methods. This is mainly because RL-based methods can dynamically adjust the control strategy according to the change in traffic flows, while the control strategy of traditional methods remains unchanged. Among all RL-based methods, dynamic duration-based methods perform better than fixed duration-based methods, mainly because the dynamic duration-based methods can improve the flexibility of the RL strategy and greatly reduce the misjudgment of the action effect caused by unreasonable settings of the phase duration.
- Both MonitorLight-r and MonitorLight-p outperform other baselines. Compared to IPDALight, MonitorLight-r achieves an improvement of 2.84% on the real-world datasets and an improvement of 5.71% on synthetic datasets. Although IPDALight dynamically adjusts the phase duration based on a heuristic algorithm, the phase duration obtained by calculating the number of vehicles on incoming lanes still cannot guarantee the best effect phase at the end of the phase. MonitorLight monitors the monitoring attribute in real-time, allowing the agent to always switch phases and achieve the best effect. Specifically, in the case of low traffic load, when the last vehicle in the queue passes the intersection, the MonitorLight agent ends the current phase in time, and reselects a new phase. In the case of high traffic flow, the MonitorLight agent analyzes the phase effect in real-time to determine whether to change the phase to maximize the traffic efficiency. It indicates that PDA-TSC has an important impact on the TSC training process, and our MonitorLight approach can effectively improve the TSC control effect.

5.6 Learning Convergence Rate Comparison

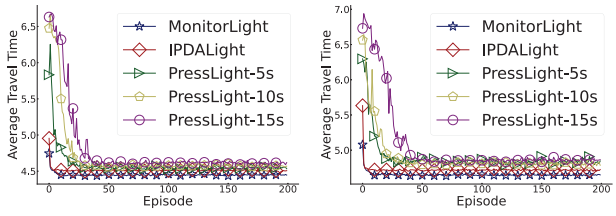
In TSC, RL methods have an essential advantage over classic methods in that RL methods can learn and adapt to dynamic traffic flow. Therefore, learning efficiency is crucial for RL methods. Higher learning efficiency can not only adapt to dynamic traffic flow conditions faster, but also reduce unnecessary computation and overhead.

Table 3: Convergence information on Syn_2×2, Syn_4×4 and Hangzhou datasets, where SEC refers to start episode # of converge, JP refers to jumpstart performance (initial performance after the first episode), and APD refers to average phase duration. The improvement in the last row refers to the degree to which MonitorLight is better than IPDALight.

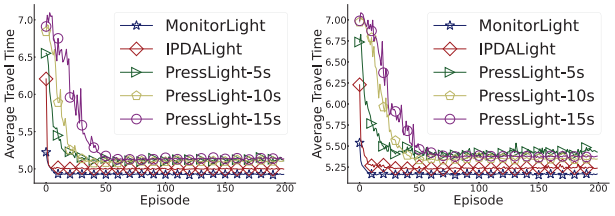
Method	Syn_2×2			Syn_4×4			Hangzhou		
	SEC	JP (s)	APD (s)	SEC	JP (s)	APD (s)	SEC	JP (s)	APD (s)
PressLight-5s	28	542.75	5	66	842.84	5	21	626.31	5
PressLight-10s	56	707.62	10	46	1068.62	10	54	770.15	10
PressLight-15s	58	837.35	15	71	1084.22	15	81	890.19	15
IPDALight	12	279.44	14.96	26	507.38	13.21	7	356.90	7.09
MonitorLight	**10**	**160.48**	10.99	**9**	**254.30**	11.09	**5**	**316.58**	14.35
Improvement	**16.67%**	**42.57%**	-	**65.38%**	**49.88%**	-	**28.57%**	**11.30%**	-

Figures 5 and 6 illustrate the training process of MonitorLight, IPDALight and PressLight on the synthetic and real-world datasets. For the sake of fairness, we set three different fixed durations (5s, 10s, 15s) for PressLight.

Figures 5 and 6 show that the dynamic duration method (MonitorLight, IPDALight) converges much faster than the fixed duration method (PressLight), implying that a reasonable phase duration plays a crucial role in the learning efficiency of the RL method in TSC. This further indicates that RL transitions collected based on dynamic duration methods have higher sample efficiency, and thus RL can converge to the optimal solution faster. Compared with other methods, MonitorLight achieves the lowest average travel time per episode, indicating that our approach has the highest learning efficiency.

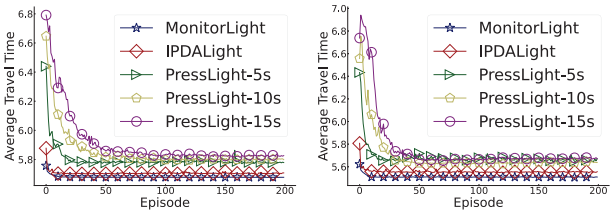


(a) Syn_1×3 (b) Syn_2×2



(c) Syn_3×3 (d) Syn_4×4

Figure 5: Convergence trends on synthetic datasets.



(a) Hangzhou (b) Jinan

Figure 6: Convergence trends on real-world datasets.

Table 3 compares PressLight, IPDALight, and MonitorLight in terms of the SEC, JP, and the average phase duration on the Syn_2×2, Syn_4×4, and Hangzhou datasets. As we can see, although PressLight exhibits a faster convergence rate with the shortening of the average phase duration, the average phase duration of the dynamic duration method is not low. This shows that the high learning efficiency of the dynamic duration method is not because of the lower average phase duration, but because the learning process is optimized by a reasonable phase duration. In addition, MonitorLight's JP is far superior to other methods. Compared to IPDALight, MonitorLight achieves a 42.57% improvement in JP on the Syn_2×2 dataset, achieving a 65.38% improvement in JP on the Syn_4×4 dataset, achieving an 11.30% improvement in JP on the Hangzhou dataset. This shows that MonitorLight's learning rate far exceeds that of other methods when encountering new traffic flow data. This further shows that our monitoring attribute can effectively represent the phase effect and solve the PDA-TSC problem.

5.7 Fairness Comparison

In the adaptive control method based on RL, due to the greedy strategy and the designed reward, most RL-based agents always aim to find optimal action in the current state. Although this control strategy has greatly improved the traffic efficiency, it also brings unfairness. Considering the intersection of the main road with high traffic flow and the alley with low traffic flow, the RL agent often selects the main road for a long time. In extreme cases, vehicles in the alley are permanently prohibited from passing. As shown in Figure 7, the west-east lane is the main road, and the traffic flow is much larger than that of the south-north lane. Therefore, the traffic light often chooses the west-east straight signal, making south-north vehicles stop in front of the traffic light for too long.

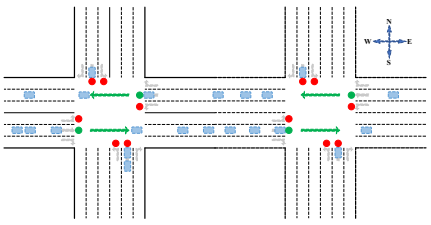


Figure 7: An illustration of the fairness problem.

Although our approach also focuses on optimizing the TSC control effect and RL learning efficiency, MonitorLight can also improve

Table 4: The fairness comparison on different datasets, where improvement over IPDALight represents the improvement of MonitorLight-f to IPDALight in the maximum waiting time.

Metric	Method	Result (s)					
		Syn_1×3	Syn_2×2	Syn_3×3	Syn_4×4	Jinan	Hangzhou
Average Travel Time	PressLight-15s	98.74	123.90	166.28	215.32	285.65	341.99
	IPDALight	88.01	109.66	146.92	184.54	255.35	298.99
	MonitorLight	84.03	103.56	136.90	173.56	246.21	292.71
	MonitorLight-f	86.70	103.74	137.76	179.23	246.28	292.74
Maximum Waiting Time	PressLight-15s	350	404	410	411	451	501
	IPDALight	207	207	260	268	223	134
	MonitorLight	207	202	248	259	96	57
	MonitorLight-f	**119**	**134**	**164**	**185**	**76**	**44**
Improvement over IPDALight		**42.51%**	**35.27%**	**36.92%**	**33.58%**	**65.92%**	**67.16%**

the issue of the strategy fairness. In line 26 of Algorithm 1, we set the greedy coefficient ε so that the agent can randomly choose the phase (i.e., any waiting vehicle has a chance to pass). More importantly, in Equation 7, the static pressure we designed considers the time the vehicle waits before the intersection. Therefore, by increasing the time coefficient ω, the agent will prioritize vehicles that have been stationary for too long to achieve the intersection fairness. We use MonitorLight and MonitorLight-f to represent the situation when $\omega = 0$ and $\omega = 0.01$ in Equation 7, respectively. Table 4 compares the control performance and fairness capability of PressLight, IPDALight, and MonitorLight on synthetic and real-world datasets, where the fairness capability is represented by the maximum waiting time (i.e., the maximum stationary time before the intersection among all vehicles, so that the lower the maximum waiting time, the better the fairness of the strategy).

As depicted in Table 4, MonitorLight and MonitorLight-f outperform other baselines in both average travel time and maximum waiting time. In terms of the fairness, the maximum waiting time of MonitorLight-f is much lower than that of other baselines. Compared to IPDALight, MonitorLight-f achieves a 66.54% improvement in maximum waiting time over real-world datasets, and obtains a 36.41% improvement on synthetic datasets. Compared with MonitorLight, the maximum waiting time of MonitorLight-f is significantly reduced, and the performance loss is minimal. This shows that considering the vehicle waiting time can effectively improve the fairness of the control strategy and cannot influence the control effect of MonitorLight very much.

5.8 Threshold Study

This subsection will explore the effect of threshold α on MonitorLight. As described in Section 4.3, to address the PDA-TSC problem, MonitorLight enables the phase to be switched in time when the optimal effect is achieved by monitoring the monitoring attribute in real-time during the training process. The threshold α, therefore, has a significant impact on the effect achieved by each phase and the frequency of phase switching.

To explore the impact of the threshold α on TSC performance, we set five different α values in the experiments conducted on four datasets. Table 5 shows the average travel time and average phase duration with different α. The results shown in Table 5 reveal that the best performance is achieved for the case of α=0.7 on synthetic datasets. As α decreases, the learning process will fail to converge. On real-world datasets, α has little effect on the performance, possibly due to the low traffic flow on the real-world datasets. In addition, α is closely related to the average phase duration. When α increases, the average phase duration gradually decreases.

Table 5: Comparison with different α.

Metric	α	Datasets			
		Syn_1×3	Syn_3×3	Jinan	Hangzhou
Average Travel Time	**0.01**	-	-	250.36	293.21
	0.1	-	-	250.67	293.20
	0.7	**84.03**	**136.90**	**246.21**	292.71
	1.5	90.87	155.17	247.61	**292.04**
	3	110.91	171.95	248.35	292.29
Average Phase Duration	**0.01**	-	-	10.21	14.46
	0.1	-	-	10.09	14.34
	0.7	9.61	9.25	8.47	14.35
	1.5	6.51	6.23	6.50	14.14
	3	5.53	5.29	5.92	13.96

6 CONCLUSION

Due to the neglect of impacts from the phase duration, existing RL-based TSC methods suffer from high average vehicle travel time and slow convergence rate. To address these issues, we formulate a novel phase-duration-aware TSC problem (PDA-TSC), and propose an RL-based TSC approach, called MonitorLight, based on the concept of mixed pressure, which considers the impact of stationary and dynamic vehicles on intersection congestion, and accurately models the intersection environment based on vehicle dynamics and waiting time. Experimental results on both real-world and synthetic datasets show that, compared with state-of-the-art methods, MonitorLight can not only reduce the average vehicle travel time, but also dramatically accelerates the RL convergence rate. In addition, our method also considers vehicle fairness, which significantly reduces the maximum vehicle waiting time.

ACKNOWLEDGMENTS

This work was supported by National Key Research and Development Program of China 2018YFB2101300, Natural Science Foundation of China 61872147, and Shanghai Trusted Industry Internet Software Collaborative Innovation Center. Mingsong Chen is the corresponding author (mschen@sei.ecnu.edu.cn).

REFERENCES

[1] Bob Pishue. 2021. Inrix 2021 traffic scorecard report. Retrieved January 10, 2022 from https://inrix.com/.

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

[3] Roger P. Roess, Elena S. Prassas, and William R. Mcshane. 2004. Traffic Engineering. *Pearson/Prentice Hall.*

[4] Saif Al-Sultan, Moath M. Al-Doori, Ali H. Al-Bayatti, and Hussien Zedan. 2014. A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications* 37 (2014), 380–392.

[5] Jianjun Shi, Xu Wu, Jizhen Guan, and Yangzhou Chen. 2013. The analysis of traffic control cyber-physical systems. *Procedia-Social and Behavioral Sciences* 96 (2013), 2487–2496.

[6] Monireh Abdoos, Nasser Mozayani, and Ana L. Bazzan. 2013. Holonic multi-agent system for traffic signals control. *Engineering Applications of Artificial Intelligence* 26, 5 (2013), 1575–1587.

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

[8] Kok-lim A. Yau, Junaid Qadir, Hooi L. Khoo, Mee H. Ling, and Peter Komisarczuk. 2017. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys* 50, 3 (2017), 1–38.

[9] Ammar Haydari and Yasin Yilmaz. 2020. Deep reinforcement learning for intelligent transportation systems: A survey. In *Proceedings of IEEE Transactions on Intelligent Transportation Systems (T-ITS).*

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602.*

[11] Tom Urbanik, Alison Tanaka, Bailey Lozner, Eric Lindstrom, Kevin Lee, Shaun Quayle, Scott Beaird, Shing Tsoi, Paul Ryus, Dong Gettman, et al. 2015. Signal timing manual. *Transportation Research Board.*

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

[13] Peter Koonce and Lee Rodegerdts. 2008. Traffic signal timing manual. *Technical report, Federal Highway Administration, United States.*

[14] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117.*

[15] 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. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM).* 1913–1922.

[16] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[17] Chang Liu, Huichu Zhang, Weinan Zhang, et al. 2020. Generalight: Improving environment generalization of traffic signal control via meta reinforcement learning. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM).* 1783-1792.

[18] Yutong Ye, Wupan Zhao, Tongquan Wei, Shiyan Hu, and Mingsong Chen. 2021. Fedlight: Federated reinforcement learning for autonomous multi-intersection traffic signal control. In *Proceedings of Design Automation Conference (DAC).* 847–852.

[19] Qize Jiang, Jingze Li, Weiwei Sun, and Baihua Zheng. 2021. Dynamic lane traffic signal control with group attention and multi-timescale reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

[20] Zhengxu Yu, Shuxian Liang, Long Wei, Zhongming Jin, Jianqiang Huang, Deng Cai, Xiaofei He, and Xiansheng Hua. 2021. Macar: urban traffic light control via active multi-agent communication and action rectification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).* 2491–2497.

[21] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the International conference on machine learning (ICML).* 1263–1272.

[22] Pravin Varaiya. 2013. The max-pressure controller for arbitrary networks of signalized intersections. *Advances in Dynamic Network Modeling in Complex Transportation Systems* 36 (2013), 27-66.

[23] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining (SIGKDD).* 1290–1298.

[24] Xiaorong Hu, Chenguang Zhao, and Gang Wang. 2020. A traffic light dynamic control algorithm with deep reinforcement learning based on gnn prediction. *arXiv preprint arXiv:2009.14627.*

[25] Salah Bouktif, Abderraouf Cheniki, and Ali Ouni. 2021. Traffic signal control using hybrid action space deep reinforcement learning. *Sensors* 21, 7 (2021), 2302.

[26] Jiechao Xiong, Qing Wang, Zhuoran Yang, Peng Sun, Lei Han, Yang Zheng, Haobo Fu, Tong Zhang, Ji Liu, and Han Liu. 2018. Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space. *arXiv preprint arXiv:1810.06394.*

[27] Wupan Zhao, Yutong Ye, Jiepin Ding, Ting Wang, Tongquan Wei, and Mingsong Chen. 2022. IPDAlight: Intensity-and phase duration-aware traffic signal control based on reinforcement learning. *Journal of Systems Architecture* 123 (2022), 102374.

[28] 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 *Proceedings of the world wide web conference (WWW).* 3620–3624.