Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control

Chacha Chen[1], Hua Wei[1], Nan Xu[2], Guanjie Zheng[1], Ming Yang, Yuanhao Xiong[4], Kai Xu[3], Zhenhui Li[1]
[1]Pennsylvania State University, [2]University of Southern California
[3]Shanghai Tianrang Intelligent Technology Co., Ltd, [4]Zhejiang University
{cjc6647,hzw77,gjz5038,jessieli}@psu.edu, nanx@usc.edu, ya0147ng@gmail.com, kai.xu@tianrang-inc.com, xiongyh@zju.edu.cn

Abstract

Traffic congestion plagues cities around the world. Recent years have witnessed an unprecedented trend in applying reinforcement learning for traffic signal control. However, the primary challenge is to control and coordinate traffic lights in large-scale urban networks. No one has ever tested RL models on a network of more than a thousand traffic lights. In this paper, we tackle the problem of multi-intersection traffic signal control, especially for large-scale networks, based on RL techniques and transportation theories. This problem is quite difficult because there are challenges such as scalability, signal coordination, data feasibility, etc. To address these challenges, we (1) design our RL agents utilizing 'pressure' concept to achieve signal coordination in region-level; (2) show that implicit coordination could be achieved by individual control agents with well-crafted reward design thus reducing the dimensionality; and (3) conduct extensive experiments on multiple scenarios, including a real-world scenario with 2510 traffic lights in Manhattan, New York City [1] [2].

Introduction

Due to the rapid urbanization, which results in an explosive increase in household owning cars, urban traffic congestion has been a significant obstacle to urbanization. Traffic congestion will not only waste fuel but also increase harmful emissions, including greenhouse gases (e.g., carbon dioxide) and other particles (e.g., nitrogen oxides) that may harm human's health (Bharadwaj et al. 2017). According to existing studies, the transport sector contributes to 23% of total CO2 emission from fuel combustion (Grote et al. 2016), and road traffic makes up about three-fourths of them. Further, for vehicles in urban cities, traffic congestion may increase the discharge by 40% (Grote et al. 2016). Therefore, mitigating traffic congestion is extremely urgent. To achieve this, one of the most effective approaches is to control the traffic signal more intelligently. Note that, the urban city is densely connected, and the signal control strategies of intersections are highly correlated. Therefore, it is crucial to solve the city-level signal control, rather than a few regions separately.

Traditional transportation approaches for traffic signal control can be categorized into following categories: pre-timed control (Koonce and Rodegerdts 2008), actuated control (Cools, Gershenson, and D'Hooghe 2013), adaptive control (Lowrie 1990; Hunt et al. 1981), and optimization-based control (Varaiya 2013). They either rely heavily on a given traffic model or depend on pre-defined rules according to expert knowledge. Hence, they fail to adjust to dynamic traffic nicely. Recently, people start to investigate reinforcement learning (RL) techniques for traffic signal control. Several studies have shown the superior performance of RL techniques over traditional transportation approaches (Wei et al. 2018; El-Tantawy and Abdulhai 2012; Van der Pol and Oliehoek 2016; Nishi et al. 2018). The biggest advantage of RL is that it directly learns how to take the next actions by observing the feedback from the environment after previous actions.

In this paper, we set out to develop a practical RL-based traffic signal control method to enable city-level traffic signal control. The method would take the traffic condition as input and learn to decide for every intersection about their next phase. We identify three key issues, that such a method must be effective to address:

• **Scalability**. Is the method able to handle large-scale traffic network? City-level traffic signal control involves thousands of traffic lights. For example, in Manhattan, NYC, there are more than about 2800 traffic signals [3]. The proposed method should be able to learn effectively on a large scale, at the same time considering the global optimization goal.

• **Coordination**. Is the method able to achieve coordination so that the global traffic conditions can be optimized? In urban environments, optimizing signal timings for traffic signals must be done jointly as signals are often in close proximity, which is commonly known as coordinating signal timings. Failure to do so can lead to decisions made at one signal deteriorating traffic operations at the other.

• **Data feasibility**. Is the method using feasible data source which makes it practical for deployment? In terms

[1] A demo video of the Manhattan experiment is provided in https://traffic-signal-control.github.io/a-thousand-lights.html

[2] Datasets and code are available at https://traffic-signal-control.github.io/

[3] https://www1.nyc.gov/html/dot/html/infrastructure/signals.shtml

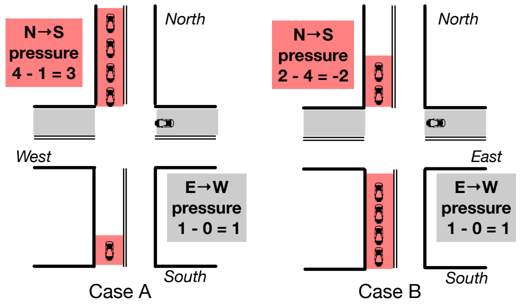


Figure 1: Illustration of max pressure control in two cases (Wei et al. 2019a). In Case A, green signal is set in the North→South direction; in Case B, green signal is set in the East→West direction.

of the deployment difficulty, the RL-based methods should not use the data that are hard to acquire in the real world.

Although there have been some existing works in using RL to control traffic signals in small regions (El-Tantawy and Abdulhai 2012; Van der Pol and Oliehoek 2016; Nishi et al. 2018), since the above three criteria cannot be met, none of the methods are applied to a city-level scenario with thousands of signals. Firstly, methods that resort to centralized optimization of coordinated agents (Prashanth and Bhatnagar 2011; Kuyer et al. 2008; Van der Pol and Oliehoek 2016) cannot satisfy *scalability*, due to the combinatorially-large joint action space. Secondly, while most methods that use decentralized RL methods can easily scale up, it is hard to assign the global-wise reward function to each intersection for *coordination*. Previous works (Nishi et al. 2018; El-Tantawy and Abdulhai 2012; Van der Pol and Oliehoek 2016) usually adopt common transportation measurements as the reward function, e.g., average wait time of vehicles (Nishi et al. 2018) and delay (El-Tantawy and Abdulhai 2012; Van der Pol and Oliehoek 2016), but there is no guarantee that the overall objective is optimized by letting each agent maximize its own expected reward. Thirdly, some RL methods assume the detailed traffic condition can be easily accessed and use complicated features to represent the traffic condition, which is unrealistic for real-world deployment. For example, aerial view about the intersection is used in (Wei et al. 2018; Van der Pol and Oliehoek 2016), while it is hard to get aerial images for every intersection in real-time, which hinders them from being deployed in the real-world application.

In this paper, we present a decentralized RL model to tackle the city-level traffic signal control problem that satisfies all the three criteria above. Specifically, we adopt the decentralized RL paradigm to enable *scalability*, upon which we further apply parameter sharing among intersections. However, naively applying parameter sharing among all the intersections will lead to inferior performance because different intersections has different structures and local traffic situations, e.g. the model of a major intersection that controls large flow can not be used to control an intersection with little traffic. Intuitively, thousands of RL agents, though controlling different traffic flows and of different structures, are essentially following similar control logic and their learned knowledge should be shared to enhance the speed of learning. To tackle this challenge, we choose FRAP (Zheng et al. 2019a), as our base model. FRAP is specifically designed to learning phase competition, the innate logic for signal control, regardless of the intersection structure and the local traffic situation. In addition, for *coordination*, we incorporate the design of RL agent with "pressure", a concept derived from max pressure control theory and aimed at maximizing the global throughput in transportation area (Varaiya 2013). Intuitively, the pressure of an intersection could be seen as the difference between upstream and downstream queue length, which indicates the inequivalence of vehicle distribution. By minimizing the pressure, our RL agent is able to balance the distribution of the vehicles within the system and maximize the system throughput. Figure 1 illustrates the concept of pressure. In detail, we design the state and the reward of our RL agent based on PressLight (Wei et al. 2019a). While the base model of the PressLight is a simple DQN network, we utilize FRAP as our base model for its generalizability to enable parameter sharing among different intersections and its superior performance. Plus, our RL model utilizes simple features like queue length (or its derivatives) that are available in real-world, which makes our model *practical*. Simulative experiments and preliminary real-world deployment shows the effectiveness of our proposed model.

In short, our contributions can be summarized as below.

- It is the first time that an intelligent traffic control algorithm is tested on a scale of thousands of traffic lights.
- We propose a decentralized network level traffic signal control RL algorithm with parameter sharing which enables large scale application.

Related Work

Conventional transportation methods for multi-intersection control usually require the intersections to have the same cycle length, and traffic of selected movements is facilitated through modifying the offset (i.e., the time interval between the beginnings of green lights) between consecutive intersections. In grid networks with homogeneous blocks, like in dense downtown areas, the coordination can be achieved by setting a fixed offset among all intersections (Urbanik et al. 2015; Roess, Prassas, and Mcshane 2011). However, few networks are so uniform for such simple treatments, which makes it difficult to provide global optimization through coordination. To solve this problem, some optimization-based methods (e.g, TRANSYT (Robertson 1969), MaxPressure (Varaiya 2013; Kouvelas et al. 2014)) are developed to optimize the global vehicle travel time, throughput, and/or the number of stops at multiple intersections (Kergaye, Stevanovic, and Martin 2010). However, such approaches still rely on assumptions to simplify the traffic condition and do not guarantee optimal results in the real world.

With the superior performance of RL-based single-intersection methods over conventional transportation methods (Wei et al. 2018; Zheng et al. 2019a; 2019b; Xiong et al. 2019), efforts have been put into developing **RL-based multi-intersection methods** (Wei et al. 2019c). For scalability concerns, *one way* is to treat all intersections isolatedly and apply individual traffic signal control methods (Mannion, Duggan, and Howley 2016; Prashanth and Bhatnagar 2011). These methods can usually be scaled up easily, but they usually cannot achieve coordination since the goal of these methods ignores neighboring intersections (El-Tantawy, Abdulhai, and Abdelgawad 2013). To achieve coordination, *an alternative way* is through centralized optimization over multiple coordinated agents in an area to ensure the optimality (Kuyer et al. 2008; Van der Pol and Oliehoek 2016; El-Tantawy, Abdulhai, and Abdelgawad 2013). However, as the network scale expands, the centralized optimization is infeasible due to the combinatorially large joint action space, which has inhibited widespread adoption of this method to a city-level control.

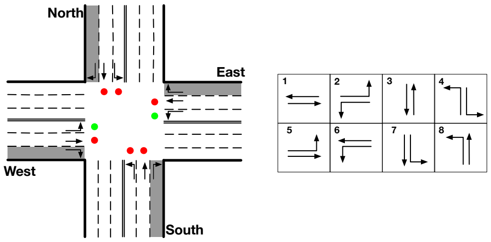
There is also a class of methods that tries to take into account both scalability and coordination with appropriate reward and state design through decentralized approaches, i.e., each agent makes decisions for its own (El-Tantawy and Abdulhai 2012; Arel et al. 2010; LIU et al. 2017; Nishi et al. 2018; Wei et al. 2019b). However, there are few of these methods that can perform well under large-scale networks due to the following issues: 1) the reward of one agent is only related to the intersection itself (e.g., minimizing the number of vehicles waiting to pass the intersection) and few reward designs are proposed for direct coordination (El-Tantawy and Abdulhai 2012; LIU et al. 2017; Nishi et al. 2018). 2) the state design of current RL-based methods usually include various features that are infeasible in the real world, e.g, cumulative delay (El-Tantawy and Abdulhai 2012; Arel et al. 2010) and/or positions of the vehicles (Van der Pol and Oliehoek 2016; LIU et al. 2017). With the above two issues, to the best of our knowledge, none of the existing RL-based methods controls a signalized network in a city level with thousands of traffic signals. Our method overcomes above two issues by utilizing simple features that are feasible in the real world and integrating the concept of "pressure" into reward design for coordination [4].

Traffic signal control systems. In many modern cities today, the widely-used adaptive traffic signal control systems such as SCATS (Lowrie 1990) and SCOOTS (Hunt et al. 1981) heavily rely on manually designed traffic signal plans. The traffic signal plans are usually generated with expert knowledge or computed by conventional traffic signal control methods. Such manually set traffic signal plans are designed to be dynamically selected according to the traffic volume detected by loop sensors. However, the loop sensors are activated only when vehicles pass through them. Thus they can only provide partial information about the vehicle through them. As a result, the signal cannot perceive and react to the real-time traffic patterns, and engineers need to manually change the traffic signal timings in the signal control system under certain traffic condition scenarios.

Preliminaries

Definition 1 (Traffic movement). *A traffic movement is defined as the traffic travelling across an intersection from one entering lane to an exiting lane. We denote a traffic movement from road l to road m as (l, m). In Figure 2 (a), there are 12 traffic movements.*

Definition 2 (Signal phase). *A traffic signal phase s is defined as a set of permissible traffic movements. As is shown in Figure 2, the intersection has eight phases with phase #2 activated. In this example, the vehicles on the left-turn lane on the East and the West are allowed to turn left to their corresponding exiting lanes. S_i denotes the set of all the phases at intersection i.*



(a) Intersection (b) Eight phases

Figure 2: The illustration of an intersection with eight phases. In this case, phase **#2** is set.

Definition 3 (Pressure of each signal phase). *For each signal phase s, there are several permissible traffic movements (l, m). Denote by $x(l, m)$ the discrepancy of the number of vehicles on lane l and lane m, for traffic movement (l, m), the pressure of a signal phase $p(s)$ is simply the total sum of the pressure of its permissable phases $\sum_{(l,m)} x(l, m), \forall(l, m) \in s$.*

Definition 4 (Pressure of an intersection). *The pressure of an intersection is the difference between the sum of the queuing vehicles on all the entering lanes and the sum of the queuing vehicles on all the exiting lanes. As is shown in Figure 3, the pressure of the middle intersection is 8.*

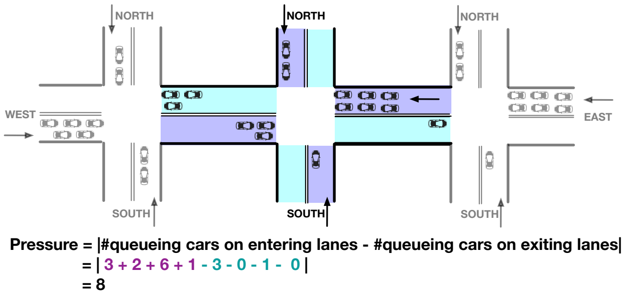

NORTH
NORTH
NORTH
WEST
EAST
SOUTH
SOUTH
SOUTH
Pressure = |#queueing cars on entering lanes - #queueing cars on exiting lanes|
= | 3 + 2 + 6 + 1 - 3 - 0 - 1 - 0 |
= 8

Figure 3: The illustration of intersection pressure.

[4] In the transportation area, methods that aim to optimize the pressure of intersections has been proved to maximize the throughput of the system under some conditions (Kouvelas et al. 2014; Varaiya 2013)

Problem 1 (Multi-intersection traffic signal control)**.** *Each intersection is controlled by an RL agent. At time step t, agent i views part of the environment as its observation o_i^t. Given the traffic situation and current traffic signal phase, the goal of the agent is to take an optimal action a (i.e., which phase to set), so that the cumulative reward r can be maximized.*

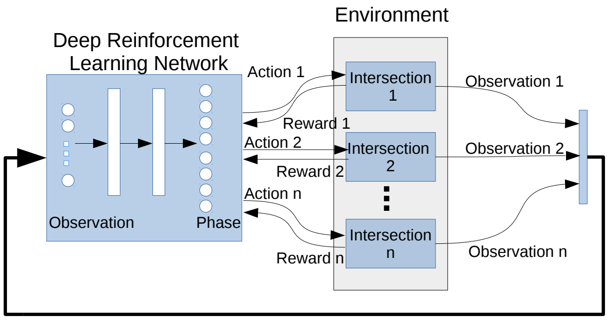


Figure 4: The framework of MPLight for multi-intersection signal control.

Method

In this section, we first introduce the concept of pressure-based control law. Then we present the proposed MPLight as a typical Deep Q-Network agent and show its learning process, as shown in Figure 4. For large-scale network signal control, we leverage parameter sharing among the agents, as discussed in the last part.

Pressure-based Coordination

For *coordination*, we incorporate into RL agent design with “pressure”, a concept derived from max pressure control theory and aimed at maximizing the global throughput in transportation area (Varaiya 2013). Intuitively, the pressure of an intersection could be seen as the difference between upstream and downstream queue length, which indicates the inequivalence of vehicle distribution. By minimizing the pressure, our RL agent is able to balance the distribution of the vehicles within the system and maximize the system throughput in return.

In previous work (Varaiya 2013), max pressure control law is proved to be stability-optimal, i.e., stabilizing and maximizing the throughput, which utilized only local information at each intersection under infinite capacities. The key idea of the max pressure control law is to set the optimization goal as minimizing the pressure for each signal phase.

Max Pressure Control Law Algorithm 1 defines max pressure control. At intersection i, for each phase $s \in S_i$, the pressure $p(s)$ is computed. Max pressure control law would select the phase with maximum pressure.

In practice, however, max pressure control is often implemented in a greedy manner, which leads to a local optimum. Hence, in the following section, we design an RL agent, PressLight, using the pressure-based reward for long-term optimization.

Algorithm 1 Max Pressure Control

```
for each intersection $i$ do
    for each phase $s$ do
        calculate $p(s)$
    end
    next phase ← $\arg\max\{p(s)|s \in S_i\}$
end
```

DQN Agent

By setting the reward of our RL agents to be the same as max pressure control objective, each local agent is maximizing its own cumulative reward, which further maximizes the network throughput under certain constraints.

- **Observation**. Each agent observes part of the system state as its own observation. For a standard intersection with 12 traffic movements, its observation includes the current phase p and the pressure of the 12 traffic movements. Note that for the intersection with fewer than 12 movements, the vector is zero-padded[5].
- **Action**. At time t, each agent chooses a phase p as its action a_t, indicating the traffic signal should be set to phase p. In this paper, agents choose from a full set of eight candidate phases, as indicated in Figure 2 . It is shown to be more flexible than providing only a subset of possible actions (Wei et al. 2019a; Zheng et al. 2019a). It should be noted that it doesn’t mean that all the eight phases must be chosen. The RL algorithm will select the best phase to set.
- **Reward**. In Figure 3, we define the reward r_i for agent i as the pressure on the intersection, which is simply the difference between the sum of the queueing vehicles on all the entering lanes and the sum of the queueing vehicles on all the exiting lanes.

If we denote the pressure of intersection i by P_i , then the reward r_i should be

$$r_i = -P_i. \tag{1}$$

By maximizing the reward, the agent is trying to stabilizing the queues in the system.

FRAP Base Model We adopt FRAP architecture as our base model. FRAP specially design a network architecture for learning the phase competition in traffic signal control problem. By modelling the phase competition relationships, FRAP has two following advantages: (1) superior performance and (2) faster training process compared with other state-of-the-art signal control methods. These two properties are especially helpful when tackling a large-scale signal control problem.

It should be noted that the idea of pressure-based design is not limited to FRAP and can also be integrated into other RL base models. As we will show in Section Ablation Study (RQ3), even by using a different base model, it is still promising to use our proposed pressure-based state and reward design.

[5] In this paper, we only consider no more than 12 traffic movements, but the proposed method can be extended to control intersections with more than 12 movements.

Deep Q-learning Following the base model, we use Deep Q-Network (DQN) to solve the multi-intersection signal control problem. Basically, our DQN takes the state features on the traffic movements as input and predicts the score (i.e., Q value) for each action candidate (i.e., phase) as described in the following Bellman Equation:

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

Parameter Sharing In Figure 4, parameters of the network are shared among all the agents. The single PressLight model receives observations from different intersections to predict the corresponding actions and learns from environment rewards for parameter update. Note that the replay memory is also shared so that all the intersections can benefit from experiences of the others.

Experiments

In this section, we conduct extensive experiments to answer the following questions:

- **RQ1**: How does our proposed method perform compared with other state-of-the-art?
- **RQ2**: Is MPLight scalable enough to control a city-level traffic signals?
- **RQ3**: What impact does the proposed techniques (e.g., pressure-based design, parameter sharing) in MPLight have on model learning?

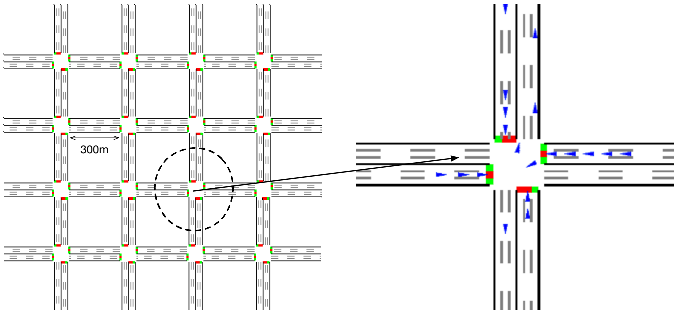


Figure 5: 4×4 road network.

Settings

Following the previous work on traffic signal control study (Wei et al. 2018), we conduct experiments on Cityflow (Zhang et al. 2019)[6]. After the traffic data being fed into the simulator, a vehicle moves towards its destination according to the setting of the environment. The simulator provides the state to the signal control method and executes the traffic signal actions from the control method. Following the tradition, each green signal is followed by a three-second yellow signal and two-second all red time to clear the intersection.

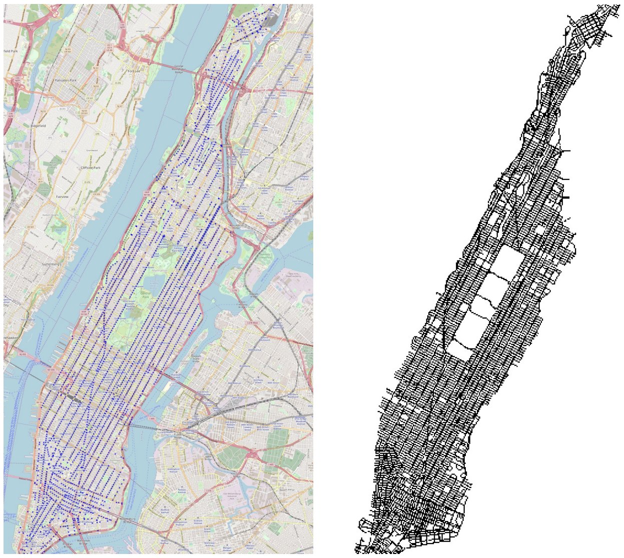
In a traffic flow dataset, each vehicle is described as (o, t, d), where o is the origin location, t is time, and d is the destination location. Locations o and d are both locations on the road network. Traffic data is fed into the simulator.

Datasets Both synthetic and real-world datasets, which focus on bi-directional and dynamical flows with turning traffic, are used in our experiments.

• Synthetic data on a 4×4 network is shown in Figure 5. Each intersection is set to be a four-way intersection, with four 300-meter long road segments. As listed in Table 1, we use four configurations to test the signal control models in different traffic demands: two types of vehicles' average arriving rate, each with Flat (0.3 variance) and Peak (0.6 variance) patterns. All the vehicles enter and leave the network at the rim edges. The turning ratios at the intersection are set as 10% (left), 60%(straight) and 30% (right) based on the statistical analysis on real-world datasets.

Table 1: Four configurations of synthetic traffic data

Config	Demand Pattern	Arrival rate (vehicles/s)
1	Flat	0.388
2	Peak	
3	Flat	0.416
4	Peak	



(a) Manhattan (b) Manhattan in simulator

Figure 6: Road network of Manhattan in our experiments.

• Real-world data. For the real-world network setting, we use the road network of Manhattan, New York City from OpenStreetMap [7] to define the network in the simulator. For traffic flow data, we use the traffic flow generated from the open-source taxi trip data[8]. We set the volume of total traffic flow as the multiplied volume of taxi data because

[6] https://cityflow-project.github.io

[7] https://www.openstreetmap.org/

[8] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

the taxi trip can be seen as a sample from real-world trip distributions. Specifically, in an hour, the total number of vehicles is approximately 25156. Note that the traffic is not uniform because it is the real-world taxi data multiplied by a scaling factor. The road network of Manhattan is demonstrated in Figure 6. Note that the Manhattan dataset contains signalized 2510 traffic lights. Vehicles are allowed to be generated and disappear at any edge in the network.

Compared Methods We compare our methods with the following baseline methods including both conventional transportation and RL methods. For a fair comparison, all the RL methods are learned without any pre-trained process and the action interval are set as 10 seconds. Each episode is a 30-minutes simulation, and we report the final results as the average of the last 10 episodes of testing.

- FixedTime (Koonce and Rodegerdts 2008): a policy gives a fixed cycle length with a predefined green ratio split among all the phases. It is widely used for steady traffic.
- MaxPressure (Varaiya 2013): the max pressure control selects the phase as green, in order to maximize the pressure according to the upstream and downstream queue length. It is the state-of-the-art control method in the transportation field for signal control in the network level.
- GRL (Van der Pol and Oliehoek 2016): a deep Q-learning algorithm for coordinated traffic signal control. Specifically, the transfer planning and the max-plus coordination algorithm are employed for large-scale coordination.
- GCN (Nishi et al. 2018): an RL-based traffic signal control method that employs a graph convolutional neural (GCN) network for representing geometric features among multiple intersections.
- PressLight (Wei et al. 2019a): a recently developed learning-based method that incorporates pressure in the state and reward design fo the RL model. It has shown superior performance in multi-intersection control problems.
- NeighborRL (Arel et al. 2010): a multi-agent deep Q-learning algorithm that feeds the model with both its own and neighbors' observations for network-level cooperation.
- FRAP (Zheng et al. 2019a): another state-of-the-art RL-based traffic signal control method with a modified network structure to capture the phase competition relation between different traffic movements.

We denote our method as MPLight, which uses the model (Q-network) structure of FRAP as the base RL model and integrate "pressure" into state and reward design.

Evaluation Metrics We select the following two representative measures to evaluate different methods.

- **Travel time.** Average travel time of all vehicles in the system is the most frequently used measure to evaluate the performance of the signal control method in transportation.
- **Throughput.** It is defined as the number of trips completed by vehicles throughout the simulation. A larger throughput in a given period means a larger number of vehicles have completed their trip during that time and indicates better control strategy.

Performance Comparison (RQ1)

We show the performance of transportation methods as well as RL models on synthetic traffic data in Table 2. The proposed MPLight consistently outperforms all the other methods in the four different scenarios, leading to both the least travel time of passengers and the maximum throughput. The maximum reduction of travel time by MPLight is 19.20% over the second optimal solution PressLight under Config 3, while the maximum enhancement of throughput over the second-best FRAP is as high as 3.08% under Config 3. The advantage of MPLight over the other transportation and reinforcement learning methods can be attributed to its decent reward design and feedback learning from the environment at the same time. Compared with the other RL methods, MPLight optimizes the control strategy by reducing the pressure between the entering and exiting lanes. Although the policy of MaxPressure also depends on the pressure of different phases, there is a large performance margin from MPLight in either travel time or throughput, since it ignores the assessment of previous actions from the environment.

Scalability Analysis (RQ2)

In this part, we turn to experiments on real-world data. We evaluate our proposed method with other baselines under Manhattan, New York City, where there are over 2500 signalized intersections. The problem of such a large scale is usually difficult to deal with through conventional methods in the transportation field.

As shown in Table 3, our method achieves the best performance on both travel time and throughput. It should be noted that two methods, GRL and NeighborRL, cannot be compared as they are unable to scale to large networks due to high complexity and computational costs. On the contrary, our proposed method MPLight can handle traffic signal control for thousands of lights effectively and efficiently.

Ablation Study (RQ3)

Impact of Pressure-based Design In this section, we test the performance of different RL-based methods with and without "pressure". It should be noted that, since PressLight already use "pressure" in its reward design, we remove the pressure design by replacing it with queue length in the reward which is similar to the reward design in FRAP.

It is shown in Table 4 that the proposed "pressure" concept could significantly boost the performance on different models, in terms of the average travel time and network throughput. The results justify the model-agnostic effectiveness of the "pressure".

The advantage of "pressure" over its base models can be attributed to its well-designed reward and state. Compared with the base models, "pressure" concept helps to optimize the control strategy by reducing the pressure between the entering and exiting lanes. Moreover, learning-based models show superior performance compared to MaxPressure because there is a large performance margin from PressLight in either travel time or throughput since it ignores the assessment of previous actions from the environment.

Table 2: Performance comparison of different methods evaluated in the four configurations of synthetic traffic data. For average travel time, the lower the better while for throughput, the higher the better.

Model	Travel Time				Throughput			
	Config 1	Config 2	Config 3	Config 4	Config 1	Config 2	Config 3	Config 4
FixedTime	573.13	564.02	536.04	563.06	3555	3477	3898	3556
MaxPressure	361.17	402.72	360.05	406.45	4702	4324	4814	4386
GRL	735.38	758.58	771.05	721.37	3122	2792	2962	2991
GCN	516.65	523.79	646.24	585.91	4275	4151	3660	3695
NeighborRL	690.87	687.27	781.24	791.44	3504	3255	2863	2537
PressLight	354.94	353.46	348.21	398.85	4887	4742	5129	5009
FRAP	340.44	298.55	361.36	598.52	5097	5113	5483	4475
MPLight	**309.33**	**262.50**	**281.34**	**353.13**	**5219**	**5213**	**5652**	**5060**

Table 3: Performance of different methods on Manhattan, a large-scale road network with 2510 traffic signals.

Model	Travel Time	Throughput
FixedTime	974.23	1940
MaxPressure	497.76	2143
GRL	-*	-*
GCN	653.45	5045
NeighborRL	-*	-*
PressLight	600.42	3447
FRAP	512.70	6346
MPLight	**472.51**	**6932**

*No result as GRL and NeighborRL can not scale up to thousands of intersections in New York's road network.

Table 4: Performance of different RL-based methods with and without "pressure" on Manhattan network.

Model	Travel Time
GCN	653.45
GCN + pressure	**646.47**
PressLight- pressure	654.04
PressLight	**600.42**
FRAP	512.70
FRAP + pressure	**472.51**

Impact of Parameter Sharing To investigate the impact of parameter sharing in model learning, we compare the performance of our RL agent design with and without parameter sharing under synthetic traffic. As is shown in Figure 7, parameter sharing enables our model to converge faster, which verifies the effectiveness of parameter sharing for controlling traffic signals.

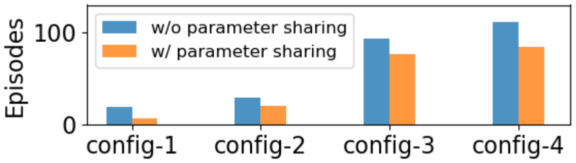


Figure 7: Number of episodes for models to converge.

Conclusion

In this paper, we propose a deep reinforcement learning method to tackle the problem of city-level traffic signal control. We are the first to evaluate the RL-based traffic signal control methods in a real-world scenario with thousands of traffic lights. Our proposed method has shown its strong performance and generalization ability.

We also acknowledge the limitations of our current approach. Although by allocating a shared agent for all intersections, the model achieves satisfactory control in the large-scale road network. However, more elaborate design for coordination and cooperation among neighboring intersections might further improve performance.

Acknowledgments

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

References

Arel, I.; Liu, C.; Urbanik, T.; and Kohls, A. 2010. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems* 4(2):128–135.

Bharadwaj, S.; Ballare, S.; Chandel, M. K.; et al. 2017. Impact of congestion on greenhouse gas emissions for road transport in mumbai metropolitan region. *Transportation research procedia* 25:3538–3551.

Cools, S.-B.; Gershenson, C.; and D'Hooghe, B. 2013. Self-organizing traffic lights: A realistic simulation. In *Advances in applied self-organizing systems*. Springer. 45–55.

El-Tantawy, S.; Abdulhai, B.; and Abdelgawad, H. 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):1140–1150.

El-Tantawy, S., and Abdulhai, B. 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*, 319–326. IEEE.

Grote, M.; Williams, I.; Preston, J.; and Kemp, S. 2016. Including congestion effects in urban road traffic co2 emissions modelling: Do local government authorities have the right options? *Transportation Research Part D: Transport and Environment* 43:95–106.

Hunt, P.; Robertson, D.; Bretherton, R.; and Winton, R. 1981. Scoot - a traffic responsive method of coordinating signals. Technical report.

Kergaye, C.; Stevanovic, A.; and Martin, P. T. 2010. Comparative evaluation of adaptive traffic control system assessments through field and microsimulation. *Journal of Intelligent Transportation Systems* 14(2):109–124.

Koonce, P., and Rodegerdts, L. 2008. Traffic signal timing manual. Technical report, United States. Federal Highway Administration.

Kouvelas, A.; Lioris, J.; Fayazi, S. A.; and Varaiya, P. 2014. Maximum Pressure Controller for Stabilizing Queues in Signalized Arterial Networks. *Transportation Research Record: Journal of the Transportation Research Board* 2421(1):133–141.

Kuyer, L.; Whiteson, S.; Bakker, B.; and Vlassis, N. 2008. Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 656–671. Springer.

LIU, M.; DENG, J.; XU, M.; ZHANG, X.; and WANG, W. 2017. Cooperative deep reinforcement learning for tra ic signal control. *The 7th International Workshop on Urban Computing (UrbComp 2018)*.

Lowrie, P. 1990. Scats: sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic.

Mannion, P.; Duggan, J.; and Howley, E. 2016. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic Road Transport Support Systems*. Springer. 47–66.

Nishi, T.; Otaki, K.; Hayakawa, K.; and Yoshimura, T. 2018. Traffic signal control based on reinforcement learning with graph convolutional neural nets. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 877–883. IEEE.

Prashanth, L. A., and Bhatnagar, S. 2011. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* 1640–1645.

Robertson, D. I. 1969. Transyt: a traffic network study tool.

Roess, R. P.; Prassas, E. S.; and Mcshane, W. R. 2011. *Traffic Engineering*. Pearson/Prentice Hall.

Urbanik, T.; Tanaka, A.; Lozner, B.; Lindstrom, E.; Lee, K.; Quayle, S.; Beaird, S.; Tsoi, S.; Ryus, P.; Gettman, D.; et al. 2015. *Signal timing manual*. Transportation Research Board.

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

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

Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496–2505. ACM.

Wei, H.; Chen, C.; Wu, K.; Xu, K.; Gayah, V.; and Li, Z. 2019a. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.

Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; and Li, Z. 2019b. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, 1913–1922. ACM.

Wei, H.; Zheng, G.; Gayah, V.; and Li, Z. 2019c. A survey on traffic signal control methods. *CoRR* abs/1904.08117.

Xiong, Y.; Zheng, G.; Xu, K.; and Li, Z. 2019. Learning traffic signal control from demonstrations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2289–2292. ACM.

Zhang, H.; Feng, S.; Liu, C.; Ding, Y.; Zhu, Y.; Zhou, Z.; Zhang, W.; Yu, Y.; Jin, H.; and Li, Z. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference*, 3620–3624. ACM.

Zheng, G.; Xiong, Y.; Zang, X.; Feng, J.; Wei, H.; Zhang, H.; Li, Y.; Xu, K.; and Li, Z. 2019a. Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, 1963–1972. ACM.

Zheng, G.; Zang, X.; Xu, N.; Wei, H.; Yu, Z.; Gayah, V.; Xu, K.; and Li, Z. 2019b. Diagnosing reinforcement learning for traffic signal control. *arXiv preprint arXiv:1905.04716*.