Deep Reinforcement Learning for Traffic Signal Control along Arterials

Hua Wei[†], Chacha Chen[‡], Kan Wu[†], Guanjie Zheng[†], Zhengyao Yu[†], Vikash Gayah[†], Zhenhui Li[†]

[†]Pennsylvania State University, [‡]Shanghai Jiao Tong Univerisity
[†]{hzw77, gjz5038, jessieli}@ist.psu.edu, [†] {kxw5389,zuy107}@psu.edu, [†] gayah@engr.psu.edu, [‡]chacha1997@sjtu.edu.cn

ABSTRACT

Arterial streets serve as the principal undertaker for urban mobility in a typical urban road network. In this paper, we propose a novel decentralized reinforcement learning method for multi-intersection traffic signal control on arterial traffic, by applying reinforcement learning control agents in each intersection. While applying individual control to multi-intersection problems faces many challenges, two main adjustments are made to optimize the overall performance: 1) to provide simple yet novel contextual information to individual agents and 2) to train the RL agents in a transfer learning way. We test our method on synthetic data dataset and show that our proposed method outperforms the state-of-the-art methods. We also interpret the policies learned by our method, which is the first time that the policy learned by the reinforcement learning control agents is interpreted using the traditional transportation coordination method on the arterial.

KEYWORDS

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

1 INTRODUCTION

Recently, reinforcement learning (RL) techniques have been applied to the real-time traffic signal control problem for a single intersection [13, 17, 20, 27, 28]; these efforts have shown that RL might provide superior performance over conventional transportation methods [6, 23]. However, in urban environments, optimizing of signal timings for adjacent 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 being made at one signal that can deteriorate traffic operations at another.

Much effort has been performed to design strategies that explicitly consider coordination between adjacent signals. This includes manually adjust offsets (i.e., time between green signal initiation at adjacent intersections) [3, 22, 26], coordinated graphs for information sharing between intersections [16, 27], or simply using a single, central agent to control all the intersections simultaneously [5]. However, explicitly coordinating in this way offers many drawbacks. Current coordinated control requires prior knowledge of traffic patterns and road length to calculate optimal offsets, road structure information to build a coordinated graph, etc. The obtained solutions are specific to the road network structure. When these features change (e.g., when new intersections are considered), the prevailing coordination policy must be created again. Centralized approaches can alleviate this but these result in large-scale optimization problems that are often not computationally feasible, especially in real-time. Information sharing approaches also rely on large communications networks to share information between signals, and such information is not feasible to obtain. Thus, a central question becomes: *can traffic signal controls operating locally with limited information sharing be able to properly coordinate operations on arterials?*

The paper proposes a method that implicitly provides coordination through a decentralized RL approach with respect to the following questions: a) for an individual intersection, what information needs to be shared with its neighbors to ensure coordination? b) How can multiple intersections learn together in a computationally efficient manner to optimize the performance of the system? To address the first question, in this paper we propose to use deep reinforcement learning (DRL) to control the traffic signal with provable state (using the distribution of the vehicle on the approaching and receiving lanes as contextual information) and reward (using queue length) design, while existing decentralized RL methods usually propose ad-hoc state and reward designs [2, 7, 8, 10, 29]. For the second question, learned knowledge is transferred from simpler to more complex arterial systems. The simpler systems serve as subproblems of the more complex systems [25] and this we can reuse knowledge optimized for the simpler systems (e.g., Q-function in RL) for each subproblem in the more complex systems without learning from scratch.

We carefully examine this method under an arterial setting with common signal control strategies that were noted in the transportation field [21]. Specifically, we compare our method with the optimal coordination strategy obtained from the transportation field, which provides a green wave in which vehicles do not have to stop while traversing the arterial. Under the synthetic data where providing a green wave is the optimal solution, the proposed RL method can achieve the same performance and automatically form a green wave, which validates our effectiveness in achieving coordination.

In summary, our contributions are as follows:

DRL4KDD '19, August 5, 2019, Anchorage, AK, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- We propose that without explicit coordination strategy, RL method through contextual information and transfer learning can also achieve the effect of coordination.
- We interpret our RL policy in connection with traditional transportation methods. Under a well-designed experiment setting where there is a closed form optimal solution that has been mathematically justified by transportation theories, we prove that our state and reward design can achieve the same optimal solution theoretically and empirically.
- We conduct experiments on both light and heavy traffic, and the results show that our method outperforms various baseline methods.

2 RELATED WORK

Individual traffic signal control. Individual traffic signal control has been investigated extensively in the field of transportation. These methods try to optimize the travel time or delay of vehicles [4, 12, 14, 24], building on the assumption that vehicles are arriving and moving in a specific pattern. Recently, reinforcement learning based methods attempt to address this problem by directly learning from the data [19, 29]. Earlier work using tabular Q-learning [1, 9] can only deal with discrete state representations. Recent work using deep Q-learning [17, 27, 28] and policy gradient [20] can cope with more complex continuous state representation and hence have shown better performance.

Conventional coordinated traffic signal control. Conventional coordinated control usually requires 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 [26]. However, few networks are so uniform for such simple treatments. It is not an easy task to even provide coordination along an arterial, given traffic of opposite directions usually cannot be facilitated simultaneously. To solve this problem, some optimization-based methods (e.g, TRANSYT [22], Maxband [18]) and traffic control systems (e.g., SCATS and SCOOT) are developed to minimize vehicle travel time and/or number of stops at multiple intersections [15]. However, such approaches still rely on assumptions to simplify the traffic condition and do not guarantee optimal results in the real world.

RL-based coordinated traffic signal control. Since recent advances in RL improve the performance on isolated traffic signal control [27, 28], efforts have been performed to design strategies that cooperate multiple RL agents. [16] and [27] consider explicit coordination mechanisms between learning agents using coordination graphs, extending [29] using the max-plus algorithm. [30] and [11] propose to use hierarchical multi-agent RL for global optimization on traffic signal control. Since all the above methods need to negotiate between the agents in the whole network, they are computationally expensive.

There is also a line of studies that use individual RL agents to control the traffic signals in the multi-intersection system [2, 7, 10]. These methods are more scalable since each agent makes its own decision based on the information from itself and neighboring intersections without explicit coordination. Our proposed method also follows this direction. However, none of the existing studies provides the theoretical justification on their state and reward design in connection with traditional transportation methods.

3 METHOD

In this section, we first present our context-aware RL design and then we discuss how to transfer knowledge between agents for more efficient learning.

3.1 Model framework: DRL agent with context

Our individual RL agent follows the state-of-the-art RL framework on a single intersection, namely *IntelliLight* [28]. The most significant difference is in the state and reward design, where we add context features on approaching and receiving lanes into state representation to realize coordination and use queue length in reward design.

Design of agent. We specify the design of three components in RL:

- **State.** Three kinds of features are included in our state: current phase P and the total number of vehicles N_i on approaching lane i which are included in [28], and a feature about contextual information - the distribution of vehicles D_i on approaching lanes. Without losing generality, in this paper, each lane i is equally binned into K segments by their relative position to the center of the intersection. Then D is a tuple $< D_{i,1}, \dots, D_{i,K} >$ with each element calculated by the number of vehicles on each segment. $k = 1$ indicates the farthest segment to the intersection and $k = K$ indicates the closest segment to the intersection. In this paper, we have $K = 3$, where each lane is binned into 3 segments. We prove that more than two segments would be efficient to describe the context, detailed in Section 3.2.
- **Action.** The action is defined as $a = 1$: change the light to next phase, and $a = 0$: keep the current phase.
- **Reward.** Our reward is defined as a sum of queue length L over all approaching lanes, where L is calculated as the total number of waiting vehicles on the given lane. A vehicle with a speed of less than 0.1 m/s is considered waiting.

Justification of state and reward design. In existing literature [13, 17, 27, 28], the state includes various kinds of features, including current phase, number of vehicles on road, delay, queue length, duration, image representation of vehicle position, etc. However, none of them theoretically justify which ones are useful. Intuitively, coordinating the traffic signals along an arterial mainly includes two decisions: a) phase split and b) offset. In our state definition, queue length helps decision (a), and the context information helps decision (b). Intuition suggests the information is sufficient for the traffic signal coordination. Going beyond intuition, we can also prove that the features we used in state definition are capable of describing the environment for an individual RL agent to learn a cooperation strategy.

Coordinating the traffic signals along arterial tries to minimize the average travel time (or equivalently delay) of vehicles under uniform traffic. We can prove that by setting reward as queue length, optimizing the reward individually is equal to optimizing the global average travel time.

3.2 Justification of state and reward definition

In this section, we will justify the choices of reward function and state features in our RL method by connecting them to the traditional transportation models.

Example: Our system is a corridor with two 4-way intersections shown in Figure 1. We examine the dynamics on the approaching lane of length l. Assume the vehicles can either run at a constant speed u or stop and wait, there is no time lost due to phase change, and there is no traffic generated within the system. For intersection j, The incoming traffic flow rate is $f_{in}^{(j)}$ vehicles per second. When there is a green light, vehicles will discharge at a rate of $f_{out}^{(j)}$ vehicles per second. The queue length is denoted as L. The vehicles have fixed routes once they enter the system.

Optimizing queue length is equal to optimizing total travel time. To understand our reward function, we consider the situation when the system described in the above example has reached a stable status. In such case, we can observe that, at the beginning of each signal cycle, the queue length will be the same. Mathematically, let t_0 be the first timestamp of a cycle, and $t_0 + C$ be the first timestamp of the next cycle, where C is the cycle length. Then we have $q_{t_0} = q_{t_0+C}$. To see why this is the case, assume the queue lengths are not the same. Then, the queue length will either increase to infinity (which violates our assumption that the system is stable) or decrease to 0 and become stable.

For a single intersection, the goal of our method is to optimize its total travel time. When there is no time loss due to phase change, the total travel time of all vehicles t_D equals to the time t_f vehicles spend on free speed u, plus the time vehicles spend on queuing t_q:

$$t_D = t_f + t_q \tag{1}$$

For t_f, once the vehicle have a fixed routing, t_f is fixed. In other words, no matter what strategy we use to control the traffic signal, t_f is a fixed period of time that vehicles have to spend within the system. In above example, t_f for a vehicle in an intersection equals to l/u.

For t_q, the total travel time of all vehicles during the cycle C equals to the integral of the number of queuing vehicles $L(t)$ with respect to time t, from t_0 to $t_0 + C$. In discrete time, we have:

$$t_q = \int_{t_0}^{t_0+C} L(t)\,dt = \sum_{t=t_0}^{C} L_t \tag{2}$$

Here, L_t is the sum of queue length over all approaching lanes at time t, approximated by the definition of our reward function.

Optimizing queue length individually is equal to optimizing the queue lengths of system when there is no spill-back. Since we adopt a decentralized control approach where each agent takes action individually, to achieve the goal of minimizing the travel time globally, we need to clarify the decision boundary of the optimality for individual level control.

For the total system with J intersections, when there is no spill-back (the phenomenon that a queue on a downstream link affects the possible output volume of the upstream link or links connected to it), the queue length L_j of an intersection j results from its own control only. Therefore, the optimization of the queue length in total system L_{all} could be treated individually:

$$\min L_{all} = \min \sum_{j}^{J} L_j = \sum_{j}^{J} \min L_j \tag{3}$$

Note that since in most real-world scenarios, the spill-back cannot be effecffively alleviated by any control strategy as the demand exceeds the capacity of the intersection. Therefore, the non-spill-back assumption is reasonable and Equation 3 holds for most cases.

Phase, queue length, contextual information are sufficient to describe the dynamic of environment. In [28] and other literature, the representation of state s includes various kinds of features, including current phase, number of vehicles on road, delay, queue length, duration, image representation of vehicle position, etc. However, the state should be chosen such that a) the agent has all the information it needs to make a good decision and b) there is little superfluous information.

Intuitively, coordinating the traffic signals along a corridor basically includes two decisions: 1) phase split and 2) offset. In state definition, phase and queue length helps decision-1, and the context information helps decision-2. So intuition suggests the features this paper uses are sufficient for the traffic signal coordination.

Going beyond intuition, we show that the dynamics of the intersection described in the example can be fully determined by those three variables, the number of vehicles N_i on the lane i and phase P, along with the contextual features D - $< D_{i,1}, \ldots, D_{i,K} >$, where $D_{i,k}$ is the number of vehicles on the road segment k of lane i ($k = 1, \ldots, K$). Here, without loss of generality, we only consider one lane i in intersection j, because the dynamics of all the lanes could be described in the same way. We denote the incoming traffic flow rate $f_{in}^{(j)}$ of intersection j, the discharging rate $f_{out}^{(j)}$, the number of vehicles N_i on lane i, the number of vehicles $D_{i,k}$ on segment k of lane i, current phase of intersection j, as f_{in}, f_{out}, N, D_k and P for simplicity.

For lane i, let $P_t = 1$ indicates that a green light is on for the lane, and 0 otherwise. Then, the transition models of the lane from transportation theory are:

$$N_{t+1} = N_t + f_{in} - f_{out} \times P_t \tag{4a}$$

$$P_{t+1} = P_t \times (1 - a_t) + (1 - P_t) \times a_t \tag{4b}$$

For segment K closest to the intersection in lane i, assuming there is no new traffic generated on the lane, we have:

$$D_{K,t+1} = D_{K,t} + D_{K-1,t-\Delta t} - f_{out} \times P_t \tag{5}$$

Above equations could be used to describe the dynamics of system states. Note that there are three variables that may not be known to the agent, f_{in} ,f_{out} and Δt, but they can be inferred from learning:

- f_{in} can be estimated by the agent from v_i at any timestamp when the red light is on: $f_{in} = v_{i,t+1} - v_{i,t}, \forall t : p_t = 0$;
- f_{out} can then be estimated when the green light is on: $f_{out} = N_{i,t} - N_{i,t+1} + f_{in}, \forall t : p_t = 1$;

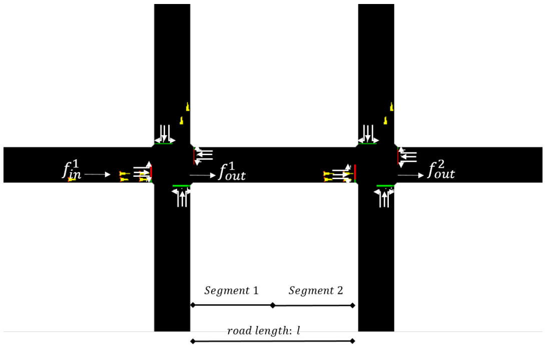


Figure 1: Exmaple road network with left and right turn

- Δt is the time a platoon of vehicles spent on traveling from segment $K-1$ to K, which can be estimated by $\Delta t = \{t_2 - t_1 \mid D_{K-1,t_1} = D_{K,t_2};\ \nabla D_{K-1,t_1} > 0,\ \nabla^2 D_{K-1,t_1} = 0;\ \nabla D_{K,t_2} > 0,\ \nabla^2 D_{K,t_2} = 0;\ t_2 > t_1\}$.

Therefore, in theory, using P, N and D as features, it is possible for the agent learn the optimal policy for the system.

3.3 Learning Process

In this paper, we adopt similar network structure as [28] for Deep Q-Network (DQN) to estimate the Q-value function as $Q(s, a; \theta)$, as shown in Figure 2. Features are concatenated and fed into fully-connected layers. Then, phase gate is used to activate different branch of the network: when phase $P = 1$, the upper branch will be activated; when phase $P = 0$, the lower branch will be activated. This will distinguish the decision process for different phases, prevent the decision from favoring specific action, and enhance the fitting ability of the network. Periodically, the agent will take samples from memory and use them to update the network as is stated in Equation 6, where a sample is a tuple of $< s, a, r, s' >$, s, a and r are the current state, action and corresponding reward, s' is the next state, and i stands for the i-th iteration.

$$\mathcal{L}_i(\theta_i) = \mathbb{E}_{s,a,r,s'}[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_{i-1})^2] \quad (6)$$

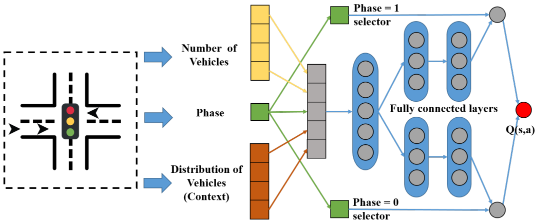


Figure 2: Q-Network structure

3.4 Transferring RL agents

In this section, we briefly introduce the process of transferring RL agents for traffic signal control problem. If we train each RL agent from scratch, the computational cost is high. Therefore, we propose to transfer knowledge, i.e., the Q-value function, between agents for a quicker convergence and better performance.

We formalize the transfer problem as follows: Given the set of RL tasks $\Sigma = \{\sigma\}$, the set of agents $\boldsymbol{G}^\sigma$ for task σ, and the mapping from each Q-value component Q to a task $M(Q) = \sigma \in \Sigma$, the transfer problem for multi-intersection signal control is to find a mapping $\mathcal{P}$ from source task σ_S to target task σ_T. We only have to define the source tasks, the corresponding mapping $\mathcal{P}_{\sigma_T}$ for each target task σ_T, and find a heuristic Q-value function Q_{σ_S} for each mapped source task.

In traffic signal control scenario, first, the agents $\boldsymbol{G}$ for simple scenarios (isolated agent or two agents) in Figure 3 are trained. Then the Q-value functions from $\boldsymbol{G}$ are re-used as the value functions Q_{σ_S} from source tasks. The corresponding mapping $\mathcal{P}_{\sigma_T}$ is simply defined by the similarity of each agent based on the topology of the traffic network.

Figure 3 shows the pipeline of transferring RL agents with bidirectional traffic on the arterial. In this process, the Q-value function is transferred. Firstly we need to train a basic RL for isolated control task O_S and then re-use it for target tasks in 2-intersection arterial (A_T, B_T). After the target task is learned, we can transfer from the 2-intersection corridor to a 3-intersection corridor (C_T, D_T, E_T), with D_T transferred from O_S, C_T and E_T transferred from A_S and B_S since their traffic flow changes similarly (e.g., both C_T and E_T have platoon traffic on WE). Similarly, after 3-intersection corridor (C_S, D_S, E_S) is learned, they can be re-used correspondingly to a 4-intersection corridor (F_T, G_T, H_T, I_T). For intersections with complex structure, transferred RL agents serves as an initialization.

4 EXPERIMENT

4.1 Experimental Setup

We configure our experimental setup using on SUMO (Simulation of Urban Mobility) [1], an open-source microscopic simulation package with flexible settings in network design, traffic simulation and traffic light control.

Road network setting. We use both synthetic and real-world road networks to define the network in the simulator. A single intersection, unless otherwise specified, is set to be a four-way intersection, with four 300-meter long road segments and six lanes with opposite directions of travel. Hence each intersection has three incoming and three outgoing lanes for each direction. The maximum speed on the road segments is set to 40 kilometers/hour. Vehicles can always turn right when there is no conflicting traffic. Every time the phase switches, a 5-second combined yellow and all-red time are followed to clear the intersection.

We have two kinds simulating environments:

(1) Homogeneous arterials with different number of four-leg intersections. Specifically, our experiments are conducted on 4 and 10 intersections. There are three approaching lanes and three receiving lanes, each 3 meters wide and 300 meters long.

(2) A heterogeneous arterial consisted of a 300-meters intersection and a 150-meters intersection. We use this environment to show the scalability of our model between heterogeneous intersections.

[1] http://sumo.dlr.de/index.html

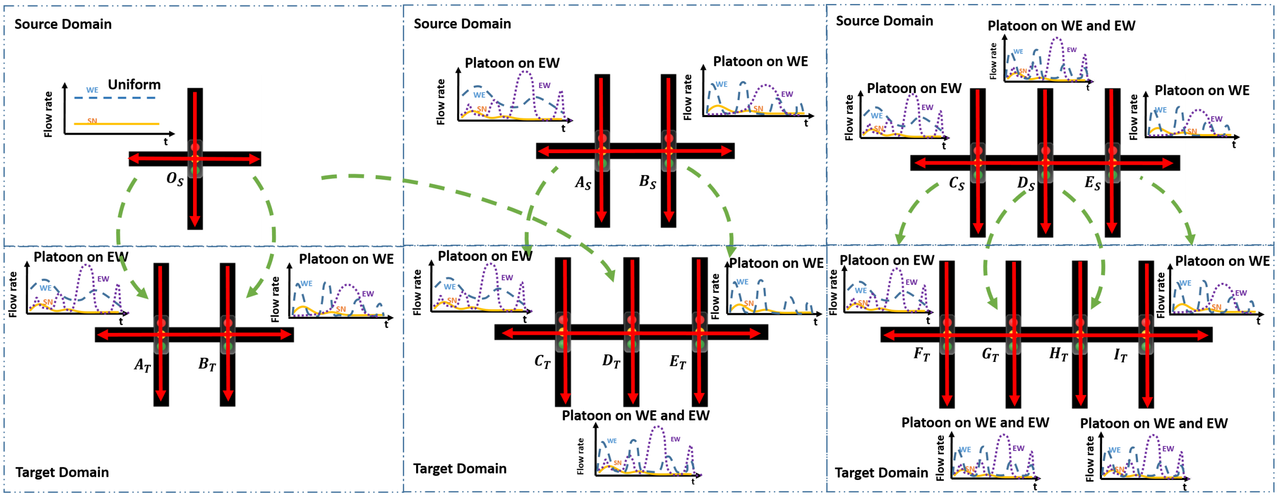


Figure 3: Transfer RL agents for multi-intersection from existing knowledge to new network. Similar agents in traffic flow network can be transferred. Left to right: Transfer from isolated intersection to a 2-intersection corridor, from 2-intersection to 3-intersection corridor, from 3-intersection to 4-intersection corridor.

The free-flow speed on the road segments in above settings is set to 40 kilometers/hour. Every time the green phase switches, a 5-second combined yellow and all-red time are followed to clear the intersection.

Evaluation metric. Following existing studies, we use the average *travel time* to evaluate the performance (other measures show similar performance and are not shown here due to space limit), which calculates average travel time the vehicles spent within the system (in seconds). This is the most frequently used measure for traffic signal performance in the transportation field.

Compared methods. We compare our model with the following methods, whose detailed configuration can be found in the cover letter. It should be noted that all methods are carefully tuned and their best results are reported.

- ***Fixedtime***: Fixed-time with random offset [23]. The offsets are randomly selected to mimic an arterial without any coordination.
- ***GreenWave*** [23] is a closed-form solution under uniform one-way traffic, providing green waves for one direction on the arterial. This is the most classical method in transportation field to implement coordination. However, *GreenWave* is only optimal in terms of average travel time for uniform one-way traffic on the arterial.
- ***Maxband*** [18] provides the green wave for two directions on the arterial.
- ***GRL*** [27] is a coordinated reinforcement learning approach for multi-intersection control by designing a coordination graph and learning the joint local Q-function for two adjacent intersections.
- ***IntelliLight*** [28] is an individual deep reinforcement learning approach. This method does not consider context information.

We denote our proposed method as ***CTRL***. Our proposed method without transfer learning is denoted as ***CRL***.

Traffic flow dataset. In the traffic flow dataset, each vehicle in the traffic flow dataset is described as (o, t, d), where o is origin location, t is time, and d is destination location. Locations o and d are both locations on the road network. Traffic data is taken as input for the simulator. Uniform traffic with turning movements is utilized as synthetic traffic with two different arrival rates on the arterial: 300 vehicles/hour/lane (light traffic) and 500 vehicles/hour/lane (heavy traffic). The arrival rates on the side streets are 30% of the arterial. All the vehicles on approaching lanes will have 20% of possibility to turn left and 20% turning right. Since all the traffic is uniform and evenly split among lanes, the demand in Table 1 is equal to critical lane volume ratios in the system, which is practically used to determine the phase split in field.

Table 1: Configurations for synthetic traffic data

Demand pattern	Arterial demand (cars/h)	Side-road demand (cars/h)
Bidirectional	300	90
	500	150

4.2 Experiment Results

4.2.1 Overall performance. Table 2 shows the average travel time performance on synthetic and real-world data. Our proposed method *CTRL* achieves the best performance compared with state-of-the-arts, including the methods with explicit coordination strategies (*GreenWave*, *Maxband* and *GRL*). This demonstrates that, even without explicit coordination, our RL agents can learn the coordination implicitly. Our proposed method consistently outperforms *IntelliLight* on both datasets. This indicates that our proposed state and reward design is effective for multi-intersection traffic signal control.

4.2.2 Efficacy of transferred knowledge. We compare a variation of our method to validate the effectiveness of transferred knowledge. In the lower part of Table 2, we can see that with transferred

Table 2: Performance of adopted methods w.r.t average travel time on a 4-intersection arterial. Lower the better.

	Synthetic light	Synthetic heavy
Transportation baselines		
Fixedtime	77.51	91.93
GreenWave	67.26	67.20
Maxband	71.20	65.93
RL baselines		
GRL	67.59	84.87
IntelliLight	65.28	68.68
Ours		
CRL	64.79	65.36
CTRL	**64.78****	**65.34****

$*$ *Maxband* is not applicable on the one-way arterial.
$**$ denotes a significance with p-value < 0.05 over the second best model (except our variant) based on a two-tailed paired t-test.

knowledge, *CTRL* consistently outperforms *CRL* under both synthetic and real-world traffic dataset in terms of average travel time, while the gap between them grows larger under real-world traffic.

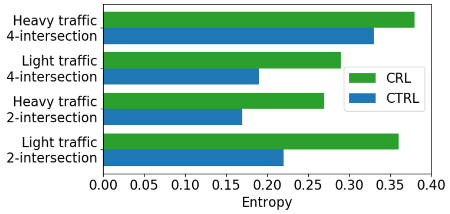


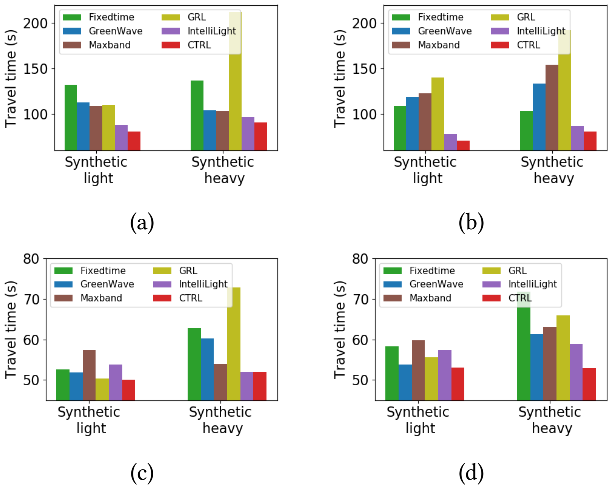
Figure 4: Cycle length entropy on traffic configurations

To further investigate the efficacy of transferred knowledge, we conduct detailed experiments under synthetic traffic. We also use *entropy of cycle length* to measure the steadiness of reinforcement learning agents. Under uniform traffic, a lower entropy indicates a more converged condition for reinforcement learning agents. Results in Figure 4 show that *CTRL* shows a lower entropy than *CRL* under both light and heavy traffic situations. This is because the knowledge transferred serves as good initialization to help the model converge faster.

4.2.3 Generality of RL agent. To test the generality of our RL method, we compare our method *CTRL* with state-of-the-art methods under following different settings:

Large scale arterial. In this experiment, an arterial with 10 homogeneous intersections is utilized to test the performance of different methods on large scale arterial. As shown in Figure 5(a), *CTRL* outperforms all other baselines under both light and heavy traffic.

Grid network. Compared with arterials, people my also concern about the overall traffic light control for an grid network. To validate the potential of our model in the grid network, we also deploy our model on a 3×3 grid network. In this network, we consider the horizontal roads as arterials. As shown in Figure 5(b), *CTRL* outperforms all other baselines under both light and heavy traffic.



(a)

(b)

(c)

(d)

Figure 5: Performance of different methods w.r.t average travel time on different road networks. (a): A 10-intersection arterial. (b): A 3×3 grid network. (c) A 2-intersection arterial with different lane length. (d) A 2-intersection arterial with different number of legs.

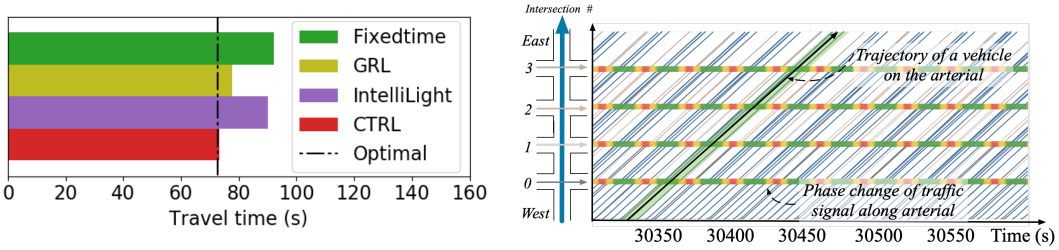
Heterogeneous intersections. We also employ our model to arterials with heterogeneous intersections. Specifically, two different kinds of heterogeneous intersections are investigated. One is an arterial with two heterogeneous intersection, where one intersection has 300-meter long roads and the other has 150-meter long roads. The other kind is a 2-intersection arterial where one intersection has four legs and the other has three legs. For intersections with different legs, we use zero-padding to fill in the missing values. As is shown in Figure 5(c) and (d), our method performs consistently better than other baselines.

4.3 Case Study: Learning Green Wave

As stated in Section 3.1, our definition of state and reward is sufficient for an RL algorithm to learn the optimal policy under uniform traffic flow. Here we validate it by using the synthetic setting where conventional coordination method *GreenWave* can form a green wave for traffic along the arterial and is an optimal solution as stated in [23]. In our experiments, the optimal offset given by *GreenWave* should be approximately 30 seconds.

Overall performance. As shown in Figure 6(a), *CTRL* outperforms all the other baseline methods and achieves almost identical performance with the optimal solution *GreenWave*.

Learned policy. We further interpret the policy learned by *CTRL* in the form of a time-space diagram, as shown in Figure 6(b). In the time-space diagram, the trajectory of a vehicle on the arterial always move from left to right and from bottom to top. Vehicles traversing green waves will always travel at their free-flow speed. As is shown in Figure 6(b), the traffic signals controlled by our method can automatically form green waves along the arterial. An exemplary green wave is highlighted as the green sloped area in Figure 6(b). A demo of the policy learned by our agent can be found at: https://bit.ly/2IQDVUv.



(a) Performances on uniform traffic

(b) Policy learned by *CTRL*

Figure 6: Overall performance and learned policy of proposed method under uniform one-way traffic. Left: Average travel time of all baseline methods. *Maxband* is not included since it does not work under one-way traffic. Right: Time-space diagram with signal timing plans to illustrate the green waves learned by *CTRL*. The green-yellow-red bands represent the change of traffic signal along the arterial. Each line in space-time diagram stands for one vehicle's trajectory.

5 CONCLUSION

In this paper, we propose a novel RL method for multi-intersection traffic signal control on the arterials with provable state and reward design. We also demonstrate the superior performance of our method over state-of-the-art methods. Specifically, we draw a connection between reinforcement learning with conventional transportation control methods.

We also acknowledge the limitations of our current approach and possible future directions could be the following. We can extend the tested arterial to the network level. The authors would expect increased computational cost while introducing more agents; however, our RL model is still elegant as there is no need for further model modifications. Also, currently our model is tested on a simulation environment, thus the feedbacks of control is also simulated. A field study is needed to validate our method in the real-world environment.

REFERENCES

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

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

[3] Bram Bakker, Shimon Whiteson, Leon Kester, and Frans CA Groen. 2010. Traffic light control by multiagent reinforcement learning systems. In *Interactive Collaborative Information Systems*. Springer, 475–510.

[4] Florence Boillot, Sophie Midenet, and Jean-Claude Pierrelee. 2006. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies* 14, 1 (2006), 18–38.

[5] Elmar Brockfeld, Robert Barlovic, Andreas Schadschneider, and Michael Schreckenberg. 2001. Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E* 64, 5 (2001), 056132.

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

[7] ALCB Bruno Castro da Silva, Denise de Oliveria, and EW Basso. 2006. Adaptive traffic control with reinforcement learning. In *Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 80–86.

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

[9] Samah El-Tantawy and Baher Abdulhai. 2010. An agent-based learning towards decentralized and coordinated traffic signal control. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (2010), 665–670. https://doi.org/10.1109/ITSC.2010.5625066

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

[11] John France and Ali A Ghorbani. 2003. A multiagent system for optimizing urban traffic. In *null*. IEEE, 411.

[12] Nathan H Gartner. 1983. *OPAC: A demand-responsive strategy for traffic signal control.* Number 906.

[13] Wade Genders and Saiedeh Razavi. 2016. Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142* (2016).

[14] Jean-Jacques Henry, Jean Loup Farges, and J Tuffal. 1984. The PRODYN real time traffic algorithm. In *Control in Transportation Systems*. Elsevier, 305–310.

[15] Cameron Kergaye, Aleksandar Stevanovic, and Peter T Martin. 2010. Comparative evaluation of adaptive traffic control system assessments through field and microsimulation. *Journal of Intelligent Transportation Systems* 14, 2 (2010), 109–124.

[16] Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. 2008. Multiagent reinforcement learning for urban traffic control using coordination graphs. *Machine learning and knowledge discovery in databases* (2008), 656–671.

[17] Li Li, Yisheng Lv, and Fei-Yue Wang. 2016. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica* 3, 3 (2016), 247–254.

[18] John DC Little, Mark D Kelson, and Nathan H Gartner. 1981. MAXBAND: A versatile program for setting signals on arteries and triangular networks. (1981).

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

[20] Seyed Sajad Mousavi, Michael Schukat, Peter Corcoran, and Enda Howley. 2017. Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning. *arXiv preprint arXiv:1704.08883* (2017).

[21] GF Newell. 1981. Blocking effects for synchronized signals. In *Proc., 8th International Symp. on Transportation and Traffic Theory, University of Toronto Press, Toronto, Canada*.

[22] Dennis I Robertson. 1969. TRANSYT: a traffic network study tool. (1969).

[23] Roger P Roess, Elena S Prassas, and William R Mcshane. 2011. *Traffic Engineering*. Pearson/Prentice Hall.

[24] Suvrajeet Sen and K Larry Head. 1997. Controlled optimization of phases at an intersection. *Transportation science* 31, 1 (1997), 5–17.

[25] Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, Jul (2009), 1633–1685.

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

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

[28] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2496–2505.

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

[30] Zhao-sheng Yang, Xin Chen, Yang-shan Tang, and Jian-ping Sun. 2005. Intelligent cooperation control of urban traffic networks. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, Vol. 3. IEEE, 1482–1486.