1. INTRODUCTION

Traffic congestion is a growing problem that continues to plague urban areas with negative outcomes to both the traveling public and society as a whole. These negative outcomes will only grow over time as more people flock to urban areas. In 2014, traffic congestion costs Americans over $166 billion in lost productivity and wasted over 3.1 billion gallons of fuel [15]. Traffic congestion was also attributed to over 56 billion pounds of harmful CO₂ emissions in 2011 [54]. Mitigating congestion would have significant economic, environmental, and societal benefits. Signalized intersections are one of the most prevalent bottleneck types in urban environments, and thus traffic signal control plays a vital role in urban traffic management.

The typical approach that transportation researchers take is to cast traffic signal control as an optimization problem under certain assumptions about the traffic model, e.g., vehicles come in a uniform and constant rate [26]. Various assumptions have to be made in order to make the optimization problem tractable. These assumptions, however, usually deviate from the real world, where the traffic condition is affected by many factors such as driver’s preference, interactions with vulnerable road users (e.g., pedestrians, cyclists, etc.), weather and road conditions. These factors can hardly be fully described in a traffic model. For a more comprehensive survey for the methods in transportation, we refer the interested readers to [27][52][40][31][7][50][64].

On the other hand, reinforcement learning methods can directly learn from the observed data without making unrealistic assumptions about the traffic model, by first taking actions to change the signal plans and then learning from the outcomes. In essence, an RL-based traffic signal control system observes the traffic condition first, then generates and executes different actions (i.e., traffic signal plans). It will then learn and adjust the strategies based on the feedback from the environment. However, in traditional RL-based methods, the states in an environment are required discretized and low-dimensional, which is one of the major limitations of the traditional approaches.

Recent advances in RL, especially deep RL, offer the opportunity to efficiently work with high dimensional input data (like images), where the agent can learn a state abstraction and a policy approximation directly from its input states. A series of related studies using deep RL for traffic signal control have appeared in the past few years. This survey is to provide an overview on the recent deep RL-based traffic signal control approaches, including the state-of-the-art methods and their experimental settings for evaluation.

In this survey, we first introduce the formulation of traffic light control problems under RL, and then classify and discuss the current RL control methods from different aspects: agent formulation, policy learning approach, and coordination strategy when facing multiple intersections. In the third section, we review how current methods are evaluated, including simulators and experimental settings that affect the performance of these methods. We then discuss some future research directions. While [39][71] provide surveys mainly on earlier studies before the popularity of deep RL, in this survey, we will mainly cover the recent deep RL methods. With the increasing interest on RL-based control mechanisms in intelligent transportation systems [24], such as autonomous driving [67] and road control [46][68], we hope this survey could also provide insights on dealing with real-world challenges for other applications in intelligent transportation systems.

2. BACKGROUND

In this section, we first describe the reinforcement learning framework which constitutes the foundation of all the methods presented in this paper. We then provide background on conventional RL-based traffic signal control, including the problem of controlling a single intersection and multiple intersections.

2.1 Reinforcement learning

Usually a single agent RL problem is modeled as a Markov Decision Process represented by \( \langle S, A, P, R, \gamma \rangle \), where their definitions are given as follows:

- Set of state representations \( S \): At time step \( t \), the agent observes state \( s^t \in S \).
The combination of deep learning with reinforcement learning helps alleviate the “curse of dimensionality” problem. Traditionally, RL is concerned with the issue of the curse of dimensionality as the number of state-action pairs can grows exponentially with the dimension of states and actions. Recent advances in deep learning helps the approximation of functions in RL like $Q(s, a)$ or $V(s)$ by learning efficiently on a significantly smaller number of features instead of a large number of state-action pairs. This helps to improve scalability with reduced requirements on memory or storage capacity, as well as reduced learning time.

3.2 Agent formulation

A key question for RL is how to formulate the RL agent, i.e., the reward, state, and action definition. In this subsection, we focus on the advances in the reward, state, and action design in recent deep RL-based methods, and refer readers interested in more detailed definitions to [16; 39; 71].

3.2.1 Reward

The choice of reward reflects the learning objective of an RL agent. In the traffic signal control problem, although the ultimate objective is to minimize the travel time of all vehicles, travel time is hard to serve as a valid reward in RL. Because the travel time of a vehicle is affected by multiple actions from traffic signals and vehicle movements, the travel time as reward would be delayed and ineffective in indicating the goodness of the signals’ action. Therefore, the existing literature often uses a surrogate reward that can be effectively measured after an action, considering factors like average queue length, average waiting time, average speed or throughput [55; 65]. The authors in [58] also take the frequency of signal changing and the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that the number of emergency stops into reward. With different reward functions being proposed, researchers in [78; 62] find out that

of the environment as state to decide its action. Various kinds of elements have been proposed to describe the environment state, such as queue length, waiting time, speed and phase, etc. These elements can be defined on the lane level or road segment level, and then concatenated as a vector. In earlier work using RL for traffic signal control, people need to discretize the state space and use a simple tabular or linear model to approximate the state functions for efficiency. However, the real-world state space is usually huge, which confines the traditional RL methods in terms of memory or performance. With advances in deep learning, deep RL methods are proposed to handle large state space as an effective function approximation. Recent studies propose to use images \[9, 14, 18, 20, 22, 23, 32, 33, 42, 58, 65\] to represent the state, where the position of vehicles are extracted as an image representation. With variant information used in state representation in different studies, \[62, 25\] shows that complex state definition and large state space do not necessarily lead to significant performance gain, and proposes to use simple state like lane-level queue length and phase to represent the environment state.

### 3.2.3 Action scheme

Now there are different types of action definitions for an RL agent in traffic signal control: (1) set current phase duration \[4, 3\], (2) set the ratio of the phase duration over pre-defined total cycle duration \[1, 10\], (3) change to the next phase in pre-defined cyclic phase sequence \[39, 48, 58, 65\], and (4) choose the phase to change to among a set of phases \[2, 9, 12, 42, 49\]. The choice of action scheme is closely related to specific settings of traffic signals. For example, if the phase sequence is required to be cyclic, then the first three action schemes should be considered, while “choosing the phase to change to among a set of phases” can generate flexible phase sequences.

### 3.3 Policy learning

RL methods can be categorized in different ways. \[3, 29\] divide current RL methods to model-based methods and model-free methods. Model-based methods try to model the transition probability among states explicitly, while model-free methods directly estimate the reward for state-action pairs and choose the action based on this. In the context of traffic signal control, the state transition between states is primarily influenced by people’s driving behaviors, which are diverse and hard to predict. Therefore, currently, most RL-based methods for traffic signal control are model-free methods. In this subsection, we take the categorization in \[43\] value-based methods and policy-based methods.

#### 3.3.1 Value-based methods

Value-based methods approximate the state-value function or state-action value function (i.e., how rewarding each state is or state-action pair is), and the policy is implicitly obtained from the learned value function. Most of the RL-based traffic signal control methods use DQN \[41\], where the model is parameterized by neural networks and takes the state representation as input \[58, 30\]. In DQN, discrete actions are required as the model directly outputs the action’s value given a state, which is especially suitable for action schema (3) and (4) mentioned in Section 3.2.3.

#### 3.3.2 Policy-based methods

Policy-based methods directly update the policy parameters (e.g., a vector of probabilities to conduct actions under specific state) towards the direction to maximizing a predefined objective (e.g., average expected return). The advantage of policy-based methods is that it does not require the action to be discrete like DQN. Also, it can learn a stochastic policy and keep exploring potentially more rewarding actions. To stabilize the training process, the actor-critic framework is widely adopted. It utilizes the strengths of both value-based and policy-based methods, with an actor controls how the agent behaves (policy-based), and the critic measures how good the conducted action is (value-based). In the traffic signal control problem, \[49\] uses DDPG \[34\] to learn a deterministic policy which directly maps states to actions, while \[4, 32, 69\] learn a stochastic policy that maps states to action probability distribution, all of which have shown excellent performance in traffic signal control problems. To further improve convergence speed for RL agents, \[51\] proposed a time-dependent baseline to reduce the variance of policy gradient updates to specifically avoid traffic jams. In the above-mentioned methods, including both value-based and policy-based methods, deep neural networks are used to approximate the value functions. Most of the literature use vanilla neural networks with their corresponding strengths. For example, Convolutional Neural Networks (CNN) are used since the state representation contains image representation \[9, 20, 22, 32, 42, 58\]. Recurrent Neural Networks (RNN) are used to capture the temporal dependency of historical states \[60\]. Special neural network structures are also proposed to incorporate prior knowledge about the states into the learning process \[65, 77\].

### 3.4 Coordination

Coordination could benefit signal control for multi-intersection scenarios. Since recent advances in RL improve the performance on isolated traffic signal control, efforts have been performed to design strategies that cooperate with multi-agent reinforcement learning (MARL) agents. Literature \[13\] categorizes MARL into two classes: Joint action learners and independent learners. Here we extend this categorization for the traffic signal control problem.

#### 3.4.1 Joint action learners

A straightforward solution is to use a single global agent to control all the intersections \[49\]. It directly takes the state as input and learns to set the joint actions of all intersections at the same time. However, these methods can result in the curse of dimensionality, which encompasses the exponential growth of the state-action space in the number of state and action dimensions. Joint action modeling methods explicitly learns to model the joint action value of multiple agents \(Q(o_1, \ldots, o_N, a)\). The joint action space grows with the increase in the number of agents to model. To alleviate this challenge, \[58\] factorizes the global Q-function as a linear combination of local subproblems, extending \[66\] using max-plus \[27\] algorithm: \(Q(o_1, \ldots, o_N, a) = \sum_{i,j} Q_{ij}(o_i, o_j, a, a_j)\), where \(i\) and \(j\) correspond to the index of neighboring agents. In other works, \[49, 12, 57\] regard the joint Q-value as a weighted sum of local Q-values, \(Q(o_1, \ldots, o_N, a) = \sum_{i,j} w_{ij} Q_{ij}(o_i, o_j, a, a_j)\), where \(w_{ij}\) is the pre-defined weights. They attempt to ensure individual agents to consider other agents’ learning process by adding a shaping term in the loss function of the individual agent’s learning process and minimizing the difference between the weighted sum of individual Q-values and the global Q-value.

#### 3.4.2 Independent learners

There is also a line of studies that use independent RL (IRL) agents to control the traffic signals, where each RL agent controls an intersection. Unlike joint action learning methods, each agent learns its control policy without knowing the reward signal of other agents. IRL without communication methods treat each intersection individually, with each agent observing its own local environment and do not use explicit communication to resolve conflicts \[39, 10, 78\].
In some simple scenarios like arterial networks, this approach has performed well with the formation of several mini green waves. However, when the environment becomes complicated, the non-stationary impacts from neighboring agents will be brought into the environment, and the learning process usually cannot converge to stationary policies if there are no communication or coordination mechanisms among agents. To deal with this challenge, the authors in [62] propose a specified reward that describes the demand for coordination between neighbors to achieve green waves. However, when the environment becomes complex, the non-stationary impacts from neighboring agents will be brought into the environment, and the learning process usually cannot converge to stationary policies if there are no communication or coordination mechanisms among agents. To deal with this challenge, the authors in [62] propose a specified reward that describes the demand for coordination between neighbors to achieve green waves.

**IRL with communication** methods enable agents to communicate between agents about their observations and behave as a group, rather than a collection of individuals in complex tasks where the environment is dynamic, and each agent has limited capabilities and visibility of the world. Typical methods directly add neighbor’s traffic condition or past actions into the observation of the ego agent, other than just using the local traffic condition of the ego agent. In this method, all the agents for different intersection share one learning model, which requires the consistent indexing of neighboring intersections. The models the influence of neighboring agents by the fixed adjacency matrix defined in Graph Convolutional Network, which indicates their assumption that the influences between neighbors is static. In other work, [63] proposes to use Graph Attentional Networks to learn the dynamic interactions between the hidden states of neighboring agents and the ego agent. It should be pointed out that there is a strong connection between methods employing max-plus to learn joint action-learners and methods using Graph Convolutional Network to learn the communication, as both of them can be seen to learn the message passing on the graph, where the former kind of methods passing the reward and the later passing the state observations.

### Table 1: Representative deep RL-based traffic signal control methods.

| Citation | Method | Cooperation | Simulator | Road net (# signals) | Traffic flow

| Value-based | With communication | Matlab | Synthetic (5) | 2,4
| Policy-based | Without communication | Aimsun | Real (50) | 5
| Policy-based | Without communication | Aimsun | Real (43) | 5
| Value-based | Without communication | CityFlow | Real (2510) | 5
| Policy-based | Joint action | SUMO | Real (30) | 4
| Value-based | - | SUMO | Synthetic (1) | 2
| Value-based | - | Paramics | Synthetic (1) | 2
| Value-based | Without communication | SUMO | Synthetic (9) | 2
| Both studied | - | SUMO | Synthetic (1) | 1
| Value-based | With communication | SUMO | Synthetic (6) | 2
| Value-based | Without communication | AIM | Synthetic (4) | 1
| Both studied | Single global | GLD | Synthetic (5) | 3
| Policy-based | - | SUMO | Real (1) | 5
| Value-based | Joint action | SUMO | Synthetic (4) | 2
| Value-based | With communication | SUMO | Real (4) | 5
| Value-based | - | SUMO | Synthetic (1) | 1,3,4,5
| Value-based | Without communication | CityFlow | Real (16) | 2,5
| Value-based | With communication | CityFlow | Real (196) | 2,5
| Value-based | Joint action | SUMO | Synthetic (36) | 1,2,3,4
| Value-based | Without communication | CityFlow | Real (16) | 3,5
| Value-based | Without communication | CityFlow | Real (5) | 4,5

---


4. **EVALUATION**

In this section, we will introduce some experimental settings that will influence the evaluation of traffic signal control strategies: evaluation metrics, simulation environment, road network setting, and traffic flow setting. A comparison of the settings that influence the evaluation are summarized in Table

4.1 **Evaluation metrics**

The objective of traffic signal control is to facilitate safe and efficient movement of vehicles at the intersection. Safety is achieved by separating conflicting movements in time and is not considered in most related literature. Various measures have been proposed to quantify efficiency of the intersection from different perspectives, including the average travel time of all vehicles, the average number of stops that vehicles experience in the network, the average queue length in the road network, and the throughput of the road network. While the performance of the same method on queue length might differ with different definitions of a “waiting” state of a vehicle, travel time and throughput are widely adopted as evaluation metrics by recent literature.

4.2 **Simulation environment**

Since deploying and testing traffic signal control strategies in the real world involves high cost and intensive labor, simulation is a useful alternative before actual implementation. Simulations of traffic signal control often involve large, heterogeneous scenarios and vehicle-level information, thus most literature relies on microscopic simulation, in which movements of individual vehicles are represented through microscopic properties such as the position and velocity of each vehicle. Some representative open-source microscopic simulators are: The Green Light District (GLD), The Autonomous Intersection Management (AIM), Simulation of Ur...
5.2 Learning efficiency
Existing RL methods for games usually require a massive number of update iterations and trial-and-errors for RL models to yield impressive results in simulated environments. These trial-and-error attempts will lead to real traffic jams in the traffic signal control problem. Therefore, how to learn efficiently is a critical question for the application of RL in traffic signal control. While there is some previous work using Meta-Learning [72] or imitation learning [69], there is still much to investigate on learning with limited data samples and efficient exploration in traffic signal control problem.

5.3 Safety issue
While RL methods learn from trial-and-error, the learning cost of RL could be critical or even fatal in the real world as the malfunction of traffic signals might lead to accidents. An open problem for RL-based traffic signal control problem is to find ways to adapt risk management to make RL agents acceptably safe in physical environments [19]. [37] directly integrates real-world constraints into the action selection process. If pedestrians are crossing the intersection, their method will not change the control actions, which can protect crossing pedestrians. However, more safety problems like handling collisions are still to be explored.

5.4 Transferring from simulation to reality
Most RL-based traffic signal control methods mainly conduct experiments in the simulator since the simulator can generate data in a cheaper and faster way than real experimentation. Discrepancies between simulation and reality confine the application of learned policies in the real world. While some work considers to learn an interpretable policy before applying to the real world [6] or to build a more realistic simulator [61; 75; 76] for direct transferring, there is still a challenge to transfer the control policies learned in simulation to reality.

6. ACKNOWLEDGMENTS
The work was supported in part by NSF awards #1652525 and #1618448, NSF Grant CMMI-1749200 and a seed grant through the Penn State Institute of CyberScience. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

7. REFERENCES


