Mobility21

A USDOT NATIONAL UNIVERSITY TRANSPORTATION CENTER

Carnegie Mellon University

Penn
University of Pennsylvania

The Ohio State University

CCAC

Alleviating Traffic Congestion: Developing and Evaluating Novel Multi-Agent Reinforcement Learning Traffic Light Coordination Techniques

PI: Fei Fang (ORCID: 0000-0003-2256-8329)

Co-PI: Norman Sadeh (ORCID: 0000-0003-4829-5533)

FINAL RESEARCH REPORT

Contract # 69A3551747111

Mobility21 Report 2023

feif

July 2023

Abstract

We have a separate cover page.

1 Introduction

Traffic congestion in cities can in part be attributed to inefficient coordination across intersections. While solutions exist that attempt to locally optimize the operation of traffic signals, coordinating these decisions across large numbers of intersections could lead to significant reductions in congestion. Effectively doing so requires however moving beyond traditional techniques and developing adaptive models that reflect the complex nature of traffic, including the behavior of diverse users such as car drivers of different types of vehicles, and possibly pedestrians. Multi-agent reinforcement learning (MARL) has shown great promise in dealing with challenging sequential decision-making problems such as train scheduling and cyber defense. Many problems in transportation systems, including traffic light control and dynamic traffic re-routing, naturally involve making a series of decisions over time to adapt to traffic conditions, which makes MARL particularly suitable to the problems. In this project, we mainly focus on the traffic signal control (TSC) problem. There have been some existing efforts in designing MARL algorithms for TSC. However, the MARL-generated TSC policies are not widely deployed in the field. We aim to better understand and address the challenges in using MARL for TSC for real-world deployment. Much work in designing algorithms for transportation has relied on simulated testbeds, yet these models have often been very coarse. To pave the way to real-world deployment, we work with Econolite, a provider of smart transportation solutions whose signal control solutions control over two-thirds of intersections nationwide.

In this project, we model the TSC problem as a Markov game, where a traffic light controller at a single intersection is modeled as an agent. These agents need to coordinate with each other to achieve global efficiency. We surveyed the key challenges in deploying MARL-based policies for TSC, and compared two commonly used traffic simulators. In addition, as part of our ongoing efforts, we set up a large-scale high-fidelity simulation environment for Strongville, Ohio, evaluate the effectiveness of the state-of-the-art MARL algorithms in the simulation environment, and develop new interpretable MARL algorithms.

2 A Review of Challenges and Opportunities in Moving Reinforcement Learning-Based Traffic Signal Control Systems Towards Reality

As the traffic volume of metropolitan areas continues to grow worldwide, gridlock is becoming an increasingly prevalent concern. According to the *2021 Urban Mobility Report* [121], gridlock led to over 4 billion hours in travel delay and $100+ million in congestion costs across the United States in 2021. This not only impacts commercial productivity but also has environmental consequences. One important mechanism for alleviating gridlock is improving the timing of traffic signals [28]. Historically, most jurisdictions have used fixed timing plans based on traffic models, which assume fixed values of factors such as lane volumes and arrival rates [183]. To minimize implementation burden, traditional traffic signal control (TSC) uses one fixed plan throughout

the entire day, or rotates through several plans based on the time of the day. However, fixed plans cannot respond in real time to changes in traffic demand [183, 54].

Large traffic volumes also offer an abundance of data that can be used for real-time optimization of signal timing plans. Many deployed systems combine logic-triggered state changes with data-driven searches over sets of schedules [183]. However, an increasing number of approaches traverse larger search spaces using optimization and scheduling algorithms [128]. Among these approaches, reinforcement learning (RL) has yielded significant improvements over fixed and actuated TSC algorithms in simulations [26]. RL allows systems to learn from the consequences of their decisions, which enables them to achieve continuous self-improvement. Deployments of RL algorithms have achieved success in a variety of complex domains involving human interaction, such as card games [21], real-time strategy games [148], and other applications in transportation such as dispatching for ride-hailing services [114].

However, to our knowledge, RL-based TSC algorithms have never been deployed. This is in spite of the fact that papers introducing novel algorithms in this area commonly list real-world deployment as a goal for future work [109]. We believe that this discrepancy has arisen due to a focus on methodological contributions, instead of on a holistic systems thinking approach based on the data-to-deployment pipeline [111]. If RL-based signal controllers are to achieve success in deployment, domain experts in TSC and in RL must have a shared view of the problem. We take a step towards bridging the gap between research and deployment by providing the first review of challenges that may arise from end-to-end deployments of RL-based TSC, which we intend to provide a common basis of collaboration between research in TSC and RL.

In this section, we begin by describing our review methodology. Then, we provide a high-level review of the fields of TSC and RL in Section, followed by a more detailed problem formulation for RL-based TSC in Section. Next, we explore four engineering challenges. For each of these challenges, we will provide a review of (1) how these challenges are significant concerns for the state of the art in RL-based TSC; (2) what practical considerations relevant to these challenges have arisen in deployments of non-RL TSC systems; and (3) what progress has been made in the RL-based TSC literature towards solving these challenges.

1. **Uncertainty in Detection.** Typically, RL-based TSC algorithms learn based on metrics such as queue length or travel time. These require accurate vehicle detection technologies, which may not always be available in the field. Strategies to deal with detector uncertainty and failure are a prerequisite of deployment.

2. **Reliability of Communications.** Some decentralization is necessary for RL-based TSC. Coordination between intersections is important for optimizing network level metrics, yet most work in RL-based TSC has not considered the practicalities of dealing with failure and latency in communications.

3. **Compliance and Interpretability.** Jurisdictions will not have confidence in RL-based signal controllers without assurances about compliance to standards (e.g., minimum green time) and safety requirements. Model interpretability is needed to ensure that signaling plans can be audited and adjusted by stakeholders.

4. **Heterogeneous Road Users.** Most simulations for RL-based TSC assume that all cars are the same size and have the same free-flow speed. However, cars share the road with pedestrians, buses, emergency vehicles, and other road users. Algorithms must detect and respond to the needs of different road users in a safe, equitable manner.

Finally, we provide concluding thoughts and suggestions for future work.

2.1 Methodology

To obtain an overview of the domain of RL-based TSC, we conducted a targeted search on Google Scholar with the keywords "traffic signal"/"traffic light", "reinforcement learning", and "review"/"survey". We identified the four challenges addressed in the following sections through these reviews. From here, we conducted snowball sampling based on their citations to locate papers in the RL literature that discuss these challenges. For RL papers, we focused on those published after 2015, since this field has rapidly evolved over the past several years. We also performed additional targeted Google Scholar searches to find literature which describes non-RL deployments of TSC, by searching the keywords "traffic signal"/"traffic light" and "adaptive" in conjunction with the following keywords:

- For challenge 1, "uncertainty", "noise", "sensing error", "accuracy'.
- For challenge 2, "coordination", "communication", "closed loop", "message", "NTCIP".
- For challenge 3, "compliance", "safety", "accountability", "interpretability" / "explainability".
- For challenge 4, "pedestrian" / "leading pedestrian interval", "cyclist", "transit", "emergency vehicle", "priority", "preempt".

2.2 Related Work

Traffic signal control (TSC) aims to allocate green time at an intersection to traffic moving in different directions. Every approach (roadway entering the intersection) is split into lanes for forward, left-turn, and (possibly) right turn movements (which may be assumed to always be permissible) [53, 182]. For efficiency, pairs of compatible movements are often arranged into phases and signaled simultaneously [109, 66, 158]. The task is to find some division of green time between phases for each intersection in a road network, which maximizes metrics such as the throughput of the network. We refer the reader to [36] for details of the problem formulation.

Different approaches to dividing green time include choosing phase durations or phase sequences, or fixing a phase sequence within a cycle and choosing the length of the cycle or the proportions of each phase within the cycle [109, 53, 158]. Three main types of algorithmic approaches exist. In fixed-time control, which has historically been a popular strategy [183], a small number of fixed plans are optimized based on

past traffic data under the assumption of uniform demand. In actuated control, detector inputs (such as vehicle presence data from loop detectors) are used in conjunction with a fixed set of logical rules. Finally, adaptive control uses more complex prediction and optimization algorithms to control signaling plans [53, 36].

One emerging approach to adaptive control has been reinforcement learning (RL). RL is a sequential decision-making paradigm wherein agents learn how to act through trial-and-error interactions with an environment. The goal of RL is to learn policies, which describe how agents should act given the state of the environment. Early work in reinforcement learning during the 1980s and 1990s, which included the seminal Q-learning algorithm [155], relied on tabular enumeration of environment states and agent actions. RL remained relatively difficult to scale until the emergence of methods based on function approximation in the 2010s, specifically the use of neural networks for deep RL [98]. Since then, the popularity and complexity of RL has experienced explosive growth. Game-playing deep RL agents have achieved superhuman performance in card games and video games with high-dimensional state and action spaces and real-time decision-making, such as AlphaGo (Go) [125], Libratus (heads-up no-limit poker) [21], and AlphaStar (StarCraft II) [148]. Deep RL has also found novel applications in practical domains such as robotics, natural language processing, finance, and healthcare [78]. Transportation has been one of the most significant applications of deep RL, with tasks including autonomous driving [63], vehicle dispatching [114] and routing [107], and traffic signal control. We refer the reader to [138] for an in-depth review of the history of reinforcement learning.

The body of work that we review in this section can be seen as a parallel to work in RL for robotics that attempts to close the gap between simulations and reality. RL methods, especially deep RL methods, require an abundance of data to learn from environmental interactions. Due to the cost of real-world data collection, simulators are often employed instead to generate large quantities of interactions. However, simulators can never perfectly emulate reality. This problem, which is referred to as the reality gap [102], has been addressed by the sim-to-real literature. Some sim-to-real methods employ randomization in sensors and controllers to learn robust policies (domain randomization); some explicitly model the reality gap and try to unify the feature spaces of the source and target environments (domain adaptation); some train policies to generalize across different tasks (meta-RL); some attempt to learn from demonstrations of behavior in target environments (imitation learning); and others attempt to improve simulators. We refer the reader to [181, 32] for surveys of these methods. In this work, we draw parallels between some of these methods and developments in RL-based TSC. However, at the same time, TSC involves unique challenges that are usually not present in robotics. Environments in robotics where sim-to-real methods have been applied (see [181]) are usually highly controlled with well-defined objectives (e.g., [7]) and minimal interaction with other agents. However, TSC may be affected by varying environmental conditions and large numbers of road users.

Various reviews of applications of RL in TSC have been published. While each of the following reviews captures distinct aspects of the field that are highly relevant to our work, none of them have focused on the key issue of practical engineering challenges that present barriers to deployment, and — crucially — how to solve them instead of leaving them as open problems.

[2], [15], and [16] provided brief syntheses of early RL-based TSC methods in reviews of applications of AI in transportation. [92] and [172] were the first to take a systematic approach to reviewing RL-based TSC algorithms; the former performed the first experimental comparison of RL algorithms with a synthetic network, while the latter addressed data sources such as models of road networks and vehicle arrivals. Both reviewed state, action, and reward formulations. These reviews considered traditional algorithms in RL such as Q-learning and SARSA.

With the increasing popularity of deep learning to address challenges of scalability in RL, [54, 115] (the latter a follow-up to [172]) both reviewed deep RL methods for TSC and provided recommendations for designing novel deep RL-based TSC algorithms. [54] focused on choosing state, action, and reward representations, with some discussion of data processing, but did not consider downstream challenges in deployment. [115] provided a broad overview of various algorithm and architecture designs with less of a focus on practicalities.

Both [158, 159] reviewed alternative state, action, and reward formulations among deep RL-based TSC algorithms, as well as options for inter-agent coordination and simulation-based evaluation. They outlined, but did not investigate, challenges to deployment. [158] further compared deep RL-based algorithms to traditional actuated and adaptive methods. Likewise, as part of a wider review on deep RL for intelligent traffic systems, [57] reviewed problem formulations and the history of algorithmic developments for RL-based TSC. Finally, [109] performed a highly systematic overview of the past 26 years of research in this domain that provides quantitative support for some of the patterns that we identify.

2.3 Reinforcement Learning-Based Traffic Signal Control

In this section, we establish the basic problem formulation for RL-based TSC. We provide only the details that are most salient to the challenges that we discuss, and we refer the reader to [54, 158, 115] for syntheses of how this general framework has been instantiated. The process involves three steps: (1) problem formulation, (2) simulation, and (3) algorithm design.

The most common sequential decision-making problem formulation for RL-based TSC is the Markov decision process (MDP), which can be formally described as a tuple (S, A, T, R, γ). In an MDP, there is a single controller agent interacting with an environment, typically one intersection, over a number of time steps $t \in \{1..T\}$. At each time step t:

State The controller receives a representation of the current intersection state, $s_t \in S$, where S is the set of all possible states. For reasons of tractability, s_t is usually some kind of abstracted, numerical representation of the intersection. As reviewed by [109], five of the most commonly included elements of state representations at the intersection level are (1) the numbers of queueing vehicles in lanes (queue length, 38%), (2) the current phase (11%), (3) the number of vehicles (10%), (4) the positions of vehicles (6%), and (5) the speeds of vehicles (6%).

Action Based on s_t, the controller chooses an action (i.e. signaling decision) $a_t \in A$ to take, where A is the set of all possible actions (assumed to be the same for each state). For a majority (62% per [109]) of works, agents choose the next phase and phase duration. In other works (32% per [109]), the action space is based on cycles, including varying the phase split and sequence in fixed-length cycles or varying the cycle length.

State Transitions The action affects the intersection immediately through a probabilistic transition to the next state, which represents the immediate effect of the signaling decision. Given s_t, a_t, each possible next state $s_{t+1} \in S$ occurs with probability

$$T(s_t, a_t, s_{t+1}) = \Pr(s_{t+1} \mid s_t, a_t)$$

where $\sum_{s_{t+1}} T(s_t, a_t, s_{t+1}) = 1, \forall s_t, a_t$. Most work in RL-based TSC adopts a model-free approach that does not model T explicitly [158], since the typical sizes of the state and action spaces make the explicit learning and representation of T prohibitively costly.

Reward After taking the action, the controller receives a reward $r_t(s_t, a_t, s_{t+1}) \in \mathbb{R}$, which is some measure of how good the signaling decision was. In TSC, this can be based on the updated state (per [109], 30% use queue lengths; 6% use vehicle counts), or on vehicle-specific quality metrics (per [109], 13% use the delays of vehicles, in terms of increase in travel time; 9% use the waiting times of vehicles; 4% use the throughput of intersections). The agent uses rewards to learn‘ how good signaling decisions are in various intersection states.

Policies The goal of the controller is to learn a policy $\pi : S \to A$ that maps the current state to the action that it should take: $a_t = \pi(s_t)$, which should be optimal in the sense that it maximizes cumulative rewards $\sum_{t=1}^{T} r_t$.

As this is a sequential problem, the agent cannot greedily choose actions to maximize estimated rewards at every time step, because its choices may have ramifications for future time steps in terms of what the state will be (e.g., clearing one intersection at time step t may cause congestion at another during time step $t+1$). The Q-value function $Q(s,a)$ encodes this notion by incorporating a decaying contribution from the expected rewards of future time steps:

$$Q(s,a) = \sum_{s'} \Pr(s' \mid s,a) \left[r(s,a,s') + \gamma \max_{a'} Q(s',a') \right]$$

where $\gamma \in [0,1]$ is a discount factor. Intuitively, the Q-value describes the value of making a signaling decision in a state, given the best decision is always made in the future. The optimal policy maximizes the Q-value:

$$\pi^*(s_t) = \text{argmax}_a Q(s_t, a)$$

MDP-based formulations for RL-based TSC make several assumptions: (1) the intersection state is fully observable; (2) state transitions always follow the same probabilistic model T given the current state s_t and signaling decision a_t; and (3) rewards accumulate additively.

Several extensions of the MDP framework have been leveraged for applications in TSC. Each of these frameworks approaches more closely to reality than MDPs, yet incurs additional computational costs. Previous reviews have not addressed these frameworks in detail.

Partially Observable MDPs In partially observable MDPs (POMDPs), the agent does not directly observe the state s_t. Instead, there is a space of observations Ω, which is usually some partial representation of the intersection state in TSC. The controller's observations $o_t \in \Omega$ are assumed to be samples from some probability distribution $O(s_t, o_t) = \Pr(o_t \mid s_t)$, with $\sum_{o_t} \Pr(o_t \mid s_t) = 1, \forall s_t$. The controller never knows exactly what state it is in; instead, it maintains a probability distribution over states known as a belief state, $b_t \in \Delta^{|S|}$, which is based on an initial state distribution $b_0 \in \Delta^{|S|}$. Thus, a POMDP can be described using a tuple $(S, A, T, R, \Omega, O, b_0, \gamma)$.

POMDPs are useful for when the same state may be observed differently by agents depending on some randomness. Works in RL-based TSC that use a POMDP framework [92, 165, 76] either use observations to represent local intersection states in a road network (e.g., [165]), or to represent incomplete inputs from detectors (as is done by [76] for connected vehicle data).

Markov Games Known either as the Markov game or the stochastic game, this framework generalizes MDPs to the setting of multi-agent RL. The key difference is that there is a set of agents $N = \{1..n\}$, and each agent has an action space A_i with $A = \prod_i A_i$. At each time step, every agent i takes its own action $a_t^{(i)}$ and its own reward $r_t^{(i)}(s_t, a_t, s_{t+1})$. Thus, a Markov game can be described using a tuple (N, S, A, T, R, γ).

Markov games are useful for network-level TSC, where there are multiple intersections to consider. Works in RL-based TSC that use a Markov game framework [158, 157] generally define each intersection's controller as an agent. This is opposed to combining all state and action spaces into a single agent, which may be difficult to scale due to high dimensionality.

In this setting, the optimal policies for controllers will always depend on knowledge of the states and actions of other controllers in the road network. It is possible for all of the policies to be learned in a centralized fashion with a single algorithm instance, or in a fully decentralized fashion where the other controllers are viewed as part of the environment. Centralized training can be computationally intensive, but decentralized training may result in the suboptimality of policies. Mechanisms for coordination can help to bridge this gap [178].

Decentralized POMDPs Decentralized POMDPs (dec-POMDPs), or partially observable Markov games, are a natural combination of POMDPs and Markov games. As applied to the TSC setting, we again assume that there are multiple intersection controllers N; in addition to the actions $a_t^{(i)}$ and rewards $r_t^{(i)}$, we also assume that each intersection controller receives its own observations $o_t^{(i)} \in \Omega_i$ with $\Omega = \prod_i \Omega_i$. Thus, a dec-POMDP can be described using a tuple $(N, S, A, T, R, \Omega, O, b_0, \gamma)$.

Although MDPs are the dominant framework in RL-based TSC, a sizeable minority of work has used dec-POMDPs. Such work [3, 79, 89, 171, 170, 62] generally defines observations as the local state for the controller's intersection, as opposed to the state of the entire road network. However, dec-POMDPs are also computationally the most difficult to solve, and the design of efficient solution methods is an area of active research [6].

It is infeasible to model the transition probabilities T explicitly for TSC. This has motivated the use of traffic simulators to generate training experiences. Non-RL methods in TSC have also extensively used simulators, which has resulted in many proprietary and open-source traffic simulators that can be used for RL applications [158].

Traffic simulators can be roughly subdivided into three groups based on the level of granularity. Microscopic simulators provide the most granular simulations, as they simulate the behavior of individual vehicles (including acceleration, deceleration, and lane changing). This behavior is generally based on car-following models, which model the acceleration of a vehicle as a time series that depends on its own speed and other vehicles' speeds. Macroscopic simulators, by contrast, aggregate vehicles into flows, with time series describing the volume, speed, and density of flows between different points in a road network. Mesoscopic simulators provide an intermediate solution between microscopic and macroscopic simulators that balances detail and computational efficiency; they may either model individual vehicles as flows, or organize groups of vehicles into platoons. We refer the reader to [13] for an overview of traffic models that have been used for these types of simulations.

In RL-based TSC, the most popular traffic simulator (as reviewed by [109]) is SUMO [69]. SUMO is an open-source microscopic simulator that can flexibly simulate various types of signaling plans, road structures, and road user types. The next most popular simulators include proprietary microscopic/mesoscopic simulators (VISSIM, PARAMICS, AIMSUN) and the open-source microscopic simulator GLD [162]. Concerns about slowness have also led to the development of dedicated simulators for RL. For instance, CityFlow [177] simplifies simulations greatly in exchange for a 20-fold speedup over SUMO. As for the contents of simulations, a majority of works (62% per [109]) have used synthetically-generated road networks and traffic flows that often have had limited complexity. However, there has been a recent shift towards using more real-world data in simulations.

Finally, an RL algorithm must be used to learn a policy. Tabular methods like Q-learning and SARSA maintain explicit representations of the Q-values for every state and action, which policies directly maximize over given the current state. However, since this is inefficient, deep RL is being increasingly used for RL-based TSC [158]. Three broad types of deep RL algorithms exist. Critic algorithms use neural networks to approximate Q-values; actor algorithms use neural networks to directly parameterize policies; and actor-critic algorithms combine these ideas by using neural network estimates of Q-values to train neural network-based policies [109]. We refer the reader to [115, 57] for more detailed reviews.

2.4 Uncertainty in Detection

2.4.1 Significance of Challenges

As noted previously, states are described in inputs to RL-based TSC algorithms using abstracted features. These include vehicles' queue lengths, positions, and speeds [109]. Many works take for granted that these features are readily available [151]. As reported by [109], 67% of surveyed papers did not envision any specific data sources. Even in papers where potential data sources were specified, it is unclear how robust the methods would be to detector noise or failure. For instance, among algorithms that use vehicle positions as state features, [46, 146, 103, 80] all used the simulator SUMO to obtain noiseless images of single-intersection toy networks; [42] extended this approach with a 3D simulator for images from the perspectives of traffic cameras; and [160] used simulated traffic in SUMO based on flow rates from traffic camera footage. Each of these methods provides a sanitized representation that may not be representative of real-world conditions. Furthermore, the loss of information to noise may cause state aliasing [130], which hinders the generalizability of learned policies to different demand scenarios [3].

2.4.2 Lessons from Deployments

Types of instruments for traffic sensing include intrusive detectors (installed into the road surface) and non-intrusive detectors (mounted above the road surface) [135, 137]. Among intrusive detectors, loop detectors are relatively inexpensive, accurate, and robust to weather and time of day, but they are also highly vulnerable to wear and tear [49]. When they fail, loop detectors are being increasingly replaced by non-intrusive detectors such as video-based and radar detection systems [135], which can be flexibly reconfigured to detect different road segments and vehicle types. However, the accuracy of these systems degrades in inclement weather, and video detectors are also inaccurate at night and on high-speed roads [137, 116]. RL-based signal controllers must be designed with these limitations in mind; learning ensembles of models [72] to capture the strengths of different detectors may improve robustness. Although data about speed and position from connected vehicles can be useful, penetration remains low, so they must be integrated with traditional detector data. [60] showed in simulations that connected vehicle data could improve adaptive control even with limited penetration. Furthermore, agencies may configure their detectors differently. To account for uncertainty in vehicle stopping positions, for instance, the size of the detection zone behind the stop bar may vary [35]; detectors may also report data at different frequencies [87]. Thus, verifying the mapping from real detector data to abstract state representations is an important task for RL-based TSC.

Agencies often address problems in detection by modifying their detection setup [135] or by configuring parameters such as passage time (i.e., the amount of time that a phase is extended for upon actuation) [137]. [128] explicitly addressed error in queue length detection for their adaptive controller SURTRAC. To mitigate underestimation, they used heuristics based on differences in vehicle counts reported by advance and stop bar detectors [167]. They considered overestimation acceptable, as it provides the

algorithm with buffer time; similarly, [23] found that moderate queue length overestimation significantly improves the performance of adaptive control.

2.4.3 Progress toward Solutions

Two lines of work in RL-based TSC have the potential to address detection uncertainty.

First, various authors have investigated the effects of reducing the dimensionality of the state space. In particular, [183] showed that complex image representations of intersection state achieve inferior performance compared to a simple representation containing only vehicle counts and phases. [156] reached similar conclusions with a state representation based on queue length. Both papers also provided optimality results that connected these formulations to traditional methods in TSC. Meanwhile, [3, 47] investigated the effects of switching to coarser state representations with a single algorithm. [47] found that occupancy and speed data (e.g., from loop detectors) yielded near-identical performance to high-fidelity position data (e.g., from cameras). However, the experiments of [3] suggested that coarser state discretizations harm generalization across sudden shifts in traffic flow. Regardless, simpler state representations could facilitate identification and debugging of issues caused by detection uncertainty.

Second, other work has attempted to imbue RL-based TSC algorithms with robustness to detection uncertainty. Several methods are analogous to domain randomization in the sim-to-real literature [7, 143]. The approach of [44] is closest to the sim-to-real literature: they randomize weather and lighting conditions in their traffic simulator and train policies based on the resulting images. [118] applied Dropout to neural network units to prevent overfitting and thus to learn robust policies. They evaluated their algorithm with a simulation of probabilistic detector failure. As is done in adversarial machine learning, [140] injected Gaussian noise into queue length observations, and validated their approach with simulations where trucks cause vehicle count overestimation. Meanwhile, to handle miscalibrated measurements, [151] combined next state prediction with imitation learning from a real traffic controller (SCOOTS), [75] used autoencoders to denoise input data, and [8] evaluated the effects of lane-blocking incidents and detector noise on performance. Finally, in a growing body of work that uses connected vehicle data for RL, [76] was the first to explicitly address partial observability by adding the phase duration into the state space to learn its impact on delay.

Overall, these methods are helpful approaches for improving the robustness of RL-based TSC to detection uncertainty. However, they should be designed and tuned to address the challenges of specific deployments, leveraging past knowledge to identify and address potential causes of detector noise or failure. It may also help to model partial observability as part of the problem, e.g. by using POMDP-based algorithms.

2.5 Reliability of Communications

2.5.1 Significance of Challenges

Some level of controller decentralization is often applied in RL-based TSC, because the computational cost of RL may be prohibitive when the state and action space dimensionalities are high. At the same time, to ensure that controllers take the traffic

conditions of other intersections into account for signaling decisions, a growing number of works have implemented mechanisms for inter-intersection coordination [159]. Typical approaches involve sharing states [89, 30, 169, 152, 176, 186], actions [45], or hidden state representations from neural networks [157, 108] between controllers for neighboring intersections. While much of this work has focused on designing neural network architectures to leverage shared information (such as graph neural networks [152, 176, 157, 108]), less attention has been devoted to the mechanisms by which information must be exchanged in the first place. If there are inconsistencies in the availability of communication infrastructure and detectors between intersections, it is unclear how they may affect the performance of RL-based TSC.

2.5.2 Lessons from Deployments

In practice, signal controllers are commonly deployed as part of closed-loop systems, where control is distributed over three levels. At the top level, traffic management centers (TMCs) make policy-based signaling decisions, often involving dialogue with other stakeholders. These decisions are used to configure field master controllers (FMCs), which are installed on-site and coordinate local intersection controllers (LICs) [27]. Each FMC aggregates traffic conditions reported by connected LICs to make signaling decisions over a small region; FMCs also synchronize the clocks of LICs to ensure that they are coordinated [53, 66]. As 90% of TSC systems in the United States are closed-loop [48], upgrades to adaptive control have largely been implemented within this hierarchical organization [87]. LICs may make some limited decisions based on local traffic conditions, but coordination is still largely delegated to FMCs even in adaptive control [27]. Transitioning to adaptive control has also required agencies to update to Type 2070 or ATC controllers [53], but some controllers in road networks may retain relatively outdated hardware [66]. RL-based signal controllers will likely be deployed into such ecosystems, where control is distributed hierarchically and different intersections have different capabilities for control and/or detection. Thus, dec-POMDP formulations and algorithms based on techniques for domain adaptation from the sim-to-real literature may be helpful.

Messages are sent between controllers and TMCs using multiple communication media in modern TSC systems [53]. For wired connections, fiber optic cables are increasingly replacing traditional copper wires or coaxial cables. Wireless communication systems implemented using radio or Wi-Fi are also becoming increasingly common [135]. Thus, communication bandwidth is not likely to be a concern, except in jurisdictions where fiber optic infrastructure is not readily available. However, a major issue reported by agencies in [135] was connection reliability: poor signal strength often results in data loss or latency. In terms of data formatting, the NTCIP 1202 standard includes standard object definitions for actuated signal controllers, which has also been used for adaptive systems [48]. Communications for RL would need to fit into this standard, at least until it is updated (as has already been done for connected vehicles) [59]. In SURTRAC, [128] encoded data for communication between neighboring intersections using JSON messages with standard types.

2.5.3 Progress toward Solutions

One line of work in RL-based TSC has sought to learn more compact representations of information. Although bandwidth is not a concern, reducing message dimensionality could still mitigate the impact of communication failures. Several algorithms directly exchange Q-values of learned policies instead of learning from exchanged state representations. In [153, 82], Q-values are directly exchanged between neighbors and weighted; [146, 171, 29] leveraged the max-plus algorithm for coordination graphs, which is known to converge to near-optimality even for cyclic graphs [64]. Meanwhile, [165] designed an architecture to exchange information from the previous time step to ensure robustness to latency, and showed that it asymptotically reduces communication relative to neighbor-based approaches by 50%. [62] demonstrated that cumulative rewards can be estimated based only on vehicle counts on inbound approaches.

Some work has also focused on designing RL-based TSC algorithms for hierarchically distributed frameworks of communication and control, which could improve RL's robustness, scalability, and applicability for deployment in closed-loop systems. [1] implemented a two-level architecture where LICs can either act independently or receive joint actions from FMCs based on predictions of the regional traffic state. [89] introduced a feudal RL algorithm, in which "manager" controllers do not directly control the actions of "worker" controllers, but instead set goals that influence their rewards. [168] trained multiple sub-policies that minimize various proxy metrics such as queue length and waiting time, and a high-level controller that adaptively delegates control to sub-policies to minimize the longer-term metric of travel time. However, all of these architectures are conceptual and further work is needed to deploy them.

2.6 Compliance and Interpretability

2.6.1 Significance of Challenges

At the heart of the fact that RL-based TSC algorithms have not been deployed are the potential regulatory and safety risks that are introduced by RL [158, 57]. The issue of trust and safety for RL is by no means exclusive to the domain of TSC [22, 41, 175], but in this case the stakes are high because controllers must interact with a large number of human users and mistakes may have fatal consequences. For RL-based signal controllers to be trusted, we need to assess — both prospectively or retrospectively — whether their decisions comply with standards and reasonable expectations [154]. However, the proliferation of deep RL algorithms based on complicated state representations runs counter to this goal, as assessment of compliance is not possible if we cannot understand or at least verify their decisions. At the same time, issues of interpretability and safety have rarely been discussed in the literature on RL-based TSC [109] and are more often mentioned as desiderata for future work in reviews [109, 158, 57].

2.6.2 Lessons from Deployments

In the real world, regulatory frameworks for traffic signaling are often scattershot. In the United States, the federal *Manual on Uniform Traffic Control Devices* [34] includes

standards about the necessity, meaning, and placement of different traffic signals. Many of these standards involve the control of individual movement signals, which would be abstracted away from RL through phase-based action space definitions. However, factors such as yellow change and red clearance intervals are left to "engineering judgement". States may impose further requirements on signal timing plans based on regional transportation policies [66]. In a review of signal timing policies for 15 states, [19] found recommendations for factors such as minimum green, yellow change, and red clearance intervals, as well as when to serve turn movements. Such recommendations should be incorporated into the design of the RL action space, as was done by [128] who treated safety constraints as inputs to SURTRAC. Yet, these recommendations can also be arbitrary and dependent on data (e.g., vehicle and pedestrian clearing times [19]), and algorithmic approaches to stakeholder preference learning [73] may help to find better values.

One common strategy to ensure the safety of signal timing plans is to review common types and causes of crashes in historical data [19]. Naturally, this is a reactive approach that requires crashes to happen in the first place, and crash reports may also be biased by severity or by environmental conditions [66]. Accident modification factors (AMFs) are a popular method of quantitative analysis; they statistically estimate the effectiveness of particular changes to signal timing plans based on their expected reductions in crash rate [86, 163, 88]. We are unaware of any work in RL that estimates or uses AMFs, but they may be a valuable pathway to interpretability. The *Highway Safety Manual* also provides standard crash risk assessment models, but these models often require extensive tuning to local conditions [136, 133, 166].

2.6.3 Progress toward Solutions

Some work has enhanced the interpretability of RL-based TSC through algorithm design. [9] focused on learning surrogate policies that are regulatable, i.e. monotonic in state variables, which allows parameters to be viewed as weights. [61] learned human-auditable decision tree surrogates using VIPER, an algorithm that identifies critical states where suboptimality harms future rewards. Closer to the interpretability literature for machine learning, [117] used SHAP values to analyze how induction loop detections contribute to choices of phases for a controller in a simulated roundabout. They found that advance detectors have higher SHAP values as they are more indicative of congestion. Similarly, [44] used Grad-CAM to generate heatmaps for image-based inputs. Instead of directly interfacing with the simulator, [104] used logical rules based on signal controllers to post-process RL policy outputs for ensuring compliance.

Further work has applied heuristic modifications to RL algorithms to enforce safety. [83] prevented their system from taking actions when pedestrians are detected in crosswalks, and enforced minimum green times for pedestrians. [38] drew on their models of rear-end conflict rates (based on various observable intersection state features [37]) to design a reward formulation that minimizes such conflicts. Similarly, [52] used a binary logistic crash risk model to define crash penalties while also minimizing waiting time. Using a state formulation based on individual signals, [81] regularized the red light duration of signaling plans to mitigate unsafe behavior caused by driver frustration with extended red lights. [174] included yellow change intervals in their action

space and added a penalty for emergency braking by vehicles.

While we have reviewed many promising methods that have been developed for the interpretability and safety of RL-based TSC, more work is still needed on determining which of these methods correspond well to stakeholder requirements. Furthermore, there is a substantial literature on safe reinforcement learning using constrained optimization [18, 33, 84], which has hitherto not been applied to TSC; it is likely that such work can provide more rigorous theoretical guarantees about algorithm behavior. We also believe that, to deal with safety failures ethically, work is needed in algorithmic accountability for RL-based signal controllers.

2.7 Heterogeneous Road Users

2.7.1 Significance of Challenges

Traditional models of traffic flow used for TSC assume, simplistically, that all vehicles are identical [20, 147]. In reality, the assumption of identical or even unimodal traffic is often unrealistic, because many types of vehicles and road users — each with different needs and behavioral patterns — interact with each other on roads. RL algorithms can still implicitly encode these assumptions through simplistic state spaces, since common state variables such as queue length and vehicle position [158] do not account for inter-vehicle variation. Although such state formulations can be helpful for deriving optimality results based on traditional models in TSC [183, 156], it is unclear how these assumptions may impact the performance and safety of RL-based signal controllers in practice, especially because road users such as pedestrians and cyclists may behave non-intuitively. Dedicated simulators developed for RL-based TSC likewise abstract away inter-vehicle variation [177]. [109] found in 160 papers on RL-based TSC that only three accounted for non-private vehicle types, and only one accounted for pedestrians.

2.7.2 Lessons from Deployments

In practice, agencies make a variety of adjustments to signaling plans to accommodate different classes of road users other than regular passenger vehicles, including pedestrians, cyclists, transit vehicles, and emergency vehicles [66]. In this section, we focus on current practice in the field for pedestrians and transit/emergency vehicles. When balancing the needs of different road user classes in RL-based signal controllers, stakeholders' requirements should be taken into account; in the US, for instance, agencies disagree on whether preemption for trains should take priority over pedestrians [19].

For pedestrians, the simplest option is for the pedestrian signal to be activated in the direction of the through movement, as is implicitly assumed by many works in RL and made explicit in some (e.g., [56]). However, doing so may cause pedestrians to impede the flow of left-turning and right-turning traffic, which creates safety hazards. In practice, leading pedestrian intervals (LePIs) mitigate this risk by allowing pedestrians to start crossing before cars are permitted to make turns [66]. Alternative phase sequence designs add lagging pedestrian intervals (after turning phases) or phases exclusively for pedestrians. [124] developed a benefit-cost model to assess the safety-delay tradeoffs

for LePIs at individual intersections. Beyond safety, additional work has tried to minimize the delay of pedestrians so that they are treated equitably compared to drivers, as codified by regulations in Germany, the UK, and China [141]. For the deployed SURTRAC system, [127] adaptively set pedestrian walk intervals based on predicted phase lengths to avoid cutting them short, while [68] considered using vehicular volumes and pedestrian actuation frequencies to switch between controller modes. We are unaware of any work in RL that has explicitly included LePIs as part of the action space formulation.

As for handling transit and emergency vehicles, typical strategies include the prioritization and preemption of signals. Prioritization handles requests made by vehicles through vehicle-to-infrastructure (V2I) communications, and may or may not result in adjustments to signaling plans. Meanwhile, preemption (often used for firetrucks or trains) deterministically replaces the signal plan with a predefined routine that favors the preempting vehicle. Typically, signal controllers need multiple cycles after preemption to recover from the interruption [66]. The adaptive SCATS controller natively implements both prioritization and preemption; compared to prior practice, [126] found that SCATS' performance improvements were robust to prioritization, and [112] found that it could reduce recovery time from preemption. These results suggest the potential of implementing prioritization and preemption with RL-based methods; in particular, explicit modelling of recovery from preemption may further improve recovery times. In addition to interactions at intersections, RL-based signal controllers should also consider the effects of transit and emergency vehicles on traffic between intersections. For instance, when buses are stopped on roads, they may block other traffic from passing. As initial steps towards implementing bus prioritization in the SURTRAC system, [91] delayed the allocation of green time in intersections located downstream from stopped buses, and [129] predicted bus dwelling times at stops using V2I communications.

2.7.3 Progress toward Solutions

One paper in RL-based TSC was cited by [109] as explicitly modelling pedestrians: [83] defined the reward using the weighted average of the local intersection's vehicular queue length, neighboring intersections' vehicular queue lengths, and the local intersection's pedestrian queue length. Beyond this paper, several other works have explicitly considered pedestrians as part of the problem formulation. [173] likewise addressed joint vehicle-pedestrian control at intersections, but made no assumptions about pedestrian detector capabilities. [180] used deep RL to control a signalized crosswalk across a road (with the actions being to set the pedestrian signal to green or red), and found that it outperformed actuation under moderate levels of pedestrian demand in simulations. [8] analyzed the performance of RL-based TSC in the presence of jaywalking pedestrians that cause vehicles to slow.

Several works in RL-based TSC have also considered prioritization and preemption. For prioritization, [44] upweighted buses and emergency vehicles in their reward formulation based on throughput; [25] used a state representation based on the cell transmission traffic model and modelled priority as a binary variable; [122] adopted an implicit approach based on minimizing delay per person instead of per vehicle; [179] and [55] both considered prioritization for trams, with the former's rewards be-

ing based on tram schedule adherence and the latter using model predictive control to model driver behavior; and [71] adaptively altered vehicles' priorities depending on queue length, waiting time, and emergency vehicle presence. For preemption, [132] learned TSC policies for emergency vehicle routing with rewards that encourage low vehicle density, and [131] used RL to learn policies for notifying connected vehicles to clear out lanes for emergency vehicles to pass.

Lastly, [104] included demand data from the field for multiple types of road users — including pedestrians, cyclists, motorcyclists, trucks, and buses — in their benchmark simulation for RL-based TSC, LemgoRL, which is based on a real road network; they also included pedestrian waiting times in rewards and enforced minimum pedestrian green times. There is a need to connect high-fidelity simulations such as LemgoRL to the various approaches for handling different road user classes that we outlined above, so as to ensure their ecological validity.

2.8 Discussion

We have reviewed four barriers to the deployment of RL-based controllers for TSC. Each of these barriers has been insufficiently addressed by the majority of new work in RL-based TSC, which has focused on algorithmic contributions. However, TSC algorithms do not exist in a vacuum — they must be trained based on data from detectors, interface with signals through controllers, and control the movements of a variety of road users. Challenges both intrinsic to RL algorithms and in other pipeline components may cascade into failures with significant implications for the efficiency and safety of transportation infrastructure. Based on our literature review, we suggested ways in which further work in RL-based TSC could address these challenges.

- **Uncertainty in detection**. Instead of assuming that state features are available without noise, considerations about detectors should be part of the algorithm design process. State spaces should be designed based on the detector types that were used to collect the data; more complex representations are not necessarily more useful. Techniques in ensemble learning and robustness for RL could deal with noise and failure, especially if multiple detector modalities are available.

- **Reliability of communications**. RL policies will likely be deployed in closed-loop signal control systems with hierarchical control architectures, where domain adaptation may be necessary. Learning concise message representations may be useful for inter-intersection communication, but it is more important to consider potential failures in communication between controllers and TMCs due to poor signal strength than restrictions on communication bandwidth.

- **Compliance and interpretability**. Flexibility to enforce different types of constraints, possibly through action and reward formulations, is necessary to ensure that RL algorithms remain compliant as traffic conditions evolve. Models for deep RL must be designed so that their output policies are easily interpretable and auditable by relevant stakeholders, and the evaluation of policies should be based not just on performance metrics but also on safety (e.g., using AMFs).

- **Heterogeneous road users**. Although simple simulations with uniform vehicles can increase RL training efficiency, the resulting policies may be suboptimal in practice. Action space formulations should be designed to incorporate signal plan modifications for different types of road users, including LePIs, prioritization, and preemption. The most important goal is to achieve equity in minimizing the delay and maximizing the safety of different road user classes.

Echoing the recommendations of [111], we emphasize the importance of engaging in consultation with agency stakeholders and experts in TSC for RL practitioners. This can break down information silos that would otherwise prevent the recognition of issues during requirements engineering and integration (cf. [105]); we could not have identified these challenges ourselves without engaging with the literature on traditional TSC. Additionally, as we discussed, the practicalities of these challenges — including the availability and configuration of detectors, signaling constraints, and the priorities of different road users — will often vary depending on the statuses of road networks and their responsible agencies. While benchmark simulations based on synthetic networks facilitate evaluation, we advocate for the creation of more simulations like [104] that incorporate realistic domain constraints. RL algorithms that are trained using such benchmarks would likely have better generalizability and robustness in deployments.

More generally, we uncovered a diversity of work that addresses each challenge, which previous reviews of TSC have not comprehensively surveyed. This suggests that RL-based TSC is closer to deployment than might be suggested by a review of state-of-the-art methods. If future developments focus on combining algorithmic improvements with both real-world considerations and reproducibility techniques to facilitate collaboration [113], we believe that the integration of RL to improve real-world transportation infrastructure is within reach.

3 Analyzing Distributional (In)Equivalence of Traffic Simulators for MARL Applications

Novel traffic signal control technologies based on reinforcement learning (RL), which learn adaptive signaling policies from simulations generated using real-world traffic data, have already achieved performance on par with and even exceeding traditional control methods [26] with in-lab experiments. However, collecting data for ITS learning remains a nontrivial task. First, many state-of-the-art deep learning approaches for ITSs inherently require large quantities of data for training [43], but they also need to be able to learn from a diverse set of experiences so that they can achieve robust performance in the real world. Second, once they are trained, ITSs will need to be able to interact with many human users, including both end-users (i.e., drivers) and decision-makers (i.e., traffic engineers and city planners). End-users have unique needs and make unique choices that must be addressed individually and equitably. Decision-makers also have complex and potentially conflicting requirements [51] that necessitate repeated iteration of the design of ITSs. Thus, data collection for ITS learning must be large-scale and continuously-occurring. However, performing data collection from

real-world transportation systems in this manner may result in significant efficiency and safety costs [43].

Due to these issues, traffic simulators are commonly used to substitute or complement real-world data collection. They serve as safe sandboxes that can generate realistic data for both ITS learning [12] and real-world decision-making [5, 40]. This work narrowly focuses on open-source traffic simulators that have been developed for the training of ITSs based on RL methods, including systems for traffic signal control, autonomous driving, ramp metering, and tolling [57]. RL algorithms are particularly demanding in terms of the number of environmental interactions that they require, so traffic simulators for RL applications must be able to quickly generate high-fidelity, high-granularity data [177].

Released in 2001, SUMO [5] is the traffic simulator that has been most commonly used to train deep RL-based ITSs [109]. It supports an expansive framework for simulation definitions, as well as efficient programmatic API access. However, it is single-threaded and thus scales less well to very large road networks. Motivated by this limitation, [177] introduced CityFlow, a traffic simulator designed for applications of RL to traffic signal control. It is multi-threaded, and consequently was reported to achieve a speedup of $>$20x over SUMO. However, its framework for simulation definitions is more restricted and focuses on aspects which are essential for RL training. Its adoption is limited but increasing [109].

If traffic simulators are to serve as training environments for RL algorithms, their modelling assumptions must be sufficiently realistic that RL-based ITSs can learn to adapt to a variety of scenarios during deployment. Thus, real-world validation is crucial. However, granular validation with real-world data is usually not possible due to the aforementioned challenges of data collection. Comparisons between simulators take a partial step towards this goal by verifying that different simulators lead to equivalent outcomes. This work compares CityFlow against the more granular SUMO. [177] included a preliminary comparison of CityFlow to SUMO using the expected travel time of vehicles in a road network. However, travel time is a long-term, system-level outcome, whereas RL algorithms use instantaneous, local features for learning. A more comprehensive comparison was performed to answer the following research questions:

RQ1 Do the low-level simulation outcomes of CityFlow and SUMO have a statistically significant level of distributional equivalence?

RQ2 How is this distributional equivalence affected by incorporating variation between vehicles in terms of different driver behavioral models?

RQ3 How is this distributional equivalence affected by the scale of the simulation, in terms of traffic demand and road network size?

3.1 Related Work

Validation is an important yet challenging aspect of the development of traffic simulators to ensure their fidelity to the real world. A multitude of road networks have been used to validate SUMO itself [17, 5] and to calibrate its car-following models [69] in comparison to detector data. For instance, [85] compared traces for vehicle counts and

crossing times between SUMO and detector data for a road network in Ingolstadt, Germany; their simulation is used in this work. Meanwhile, CityFlow was validated by comparing average travel time under various traffic volumes to that of SUMO [177].

A number of studies have compared outcomes from multiple simulators; this work falls in this setting. [90] compared vehicle counts under different traffic demand and driver behavior settings for SUMO, the commercial simulator VISSIM, and TRANSIMS. Several studies involved SUMO and the commercial simulator AIMSUN. [74] compared travel times and queue lengths in these simulators under different traffic management interventions for a roundabout in Norrköping, Sweden. [119] used t-tests to compare flows and speeds in these simulators to observations from a highway interchange in Stockholm, Sweden. [14] applied t-tests to vehicle counts from the two simulators for a road network from Bucaramanga, Colombia. This work differs from prior approaches in that: (1) the measures used relate to the distribution of outcomes across the network, not just at individual points; and (2) these measures were evaluated across multiple road network scales.

There has been a significant body of literature on building realistic models of driver behavior. Car-following behavior was the first to be modelled, with the early Gazis-Herman-Rothery model being over 50 years old. Subsequent work has yielded optimal velocity, fuzzy logic, collision avoidance, action point, and cellular automaton models [77, 120]. Collision avoidance and action point models were implemented for this work. Lane-changing behavior has been modelled by a newer, separate line of work [100, 185], which has produced models based on rules, discrete choice, game theory, and cellular automata. Some work has built unified models of car-following, lane-changing, and other driver behavior [145, 93]. Here, car-following and lane-changing models were considered separately but co-varied in experiments to elucidate the effects of their interactions. The most relevant prior work is [24], who analyzed the impact of driver behavior on system-level travel time in SUMO; by contrast, this work focuses on distributional equivalence between two simulators in terms of lower-level outcomes.

3.2 Traffic Simulators for RL

Now we review the use of traffic simulators for RL-based ITS training.

RL algorithms learn policies to maximize their expected cumulative reward through repeated interactions with an environment, which for ITSs is the traffic simulator. Examples of cumulative reward objectives include travel time minimization for traffic signal control and speed limit control [109, 164], and revenue maximization for tolling [110]. However, cumulative rewards are not useful for learning because they are system-level measures calculated over longer timespans. Instead, RL algorithms observe instantaneous vehicle-level state features, use them to select actions, and then receive instantaneous rewards [57].

Formulations of the observation and reward spaces vary between papers, but they are rarely aligned directly with the cumulative reward. Observations usually consist of counts (e.g., queue lengths), positions, speeds, or other aggregated properties of individual vehicles [109, 110, 164]. Rewards are more problem-dependent. For traffic signal control, queue length, waiting time, and speed are common factors [65]; speed limit control may use proxies for travel time, crash probability, or emissions [164];

and tolling may optimize a combination of vehicle count, travel time on segments, and collected tolls [110]. Regardless of these different formulations, it is evident that system-level measures involving the movement of vehicles through the whole road network are not directly used for RL training.

Microscopic traffic simulators are most often used for RL in transportation [57]. In contrast to macroscopic and mesoscopic simulators, microscopic simulators are primarily agent-based, as they model the behavior of individual vehicles [13]. This is helpful for gathering the aforementioned types of vehicle-level observations. Both CityFlow and SUMO are microscopic. They represent road networks as graphs, with intersections as nodes ("junctions" in SUMO) and roads as edges between nodes. Vehicles are generated by flows; all vehicles within a flow share similar attributes, including behavioral parameters and routes. Routes are defined as sequences of edges. Where edges are joined at an intersection, vehicles select a pair of lanes ("roadlinks" in CityFlow, "connections" in SUMO) to traverse the intersection. However, key differences exist in finer details. Unlike CityFlow, SUMO supports definitions for more complex types of lanes (e.g., sidewalks, bicycle lanes, and ramps) and vehicle classes (e.g., public transit and emergency vehicles, pedestrians, and cyclists). It also supports stochastic generation of routes and vehicle types, and more sophisticated traffic signaling including yellow lights.

Where possible, the virtual experiments in this work were designed to control or co-vary differences between the simulators. In particular, flows in both CityFlow and SUMO were converted to be single-vehicle and deterministic, and to share the same routes. However, differences remain, particularly aspects of driver behavior driven by random perception error in SUMO; by contrast, CityFlow only uses random generation for vehicle priority. Such uncontrolled factors may account for their differences.

3.3 Varying Driver Behavior

Addressing RQ2 requires distributional equivalence to be assessed under different variations of driver behavior; as established in Section 3.1, these are implemented through car-following and lane-changing models in traffic simulators. In both CityFlow and SUMO, car-following logic is used to maintain a safe gap to the leading vehicle, while lane changes are used to switch vehicles between lanes so that they can take appropriate roadlinks at intersections to continue their routes. However, in CityFlow, car-following and lane-changing are handled in separate threads; in SUMO, they are handled in sequence, with car-following logic being executed before lane-changing logic.

3.3.1 Car-Following Models

Car-following models determine the speeds at which vehicles travel unobstructed (free speed), follow behind a lead vehicle (following speed), and stop at an obstacle (stopping speed). Two types of numerical integration can be used to calculate vehicle speeds: a Euler update, which solves for the speed at discrete timesteps, and a ballistic update, which solves for the acceleration at discrete timesteps and applies it to the speed. Both simulators were controlled to use ballistic updates.

Table 1: Parameter settings for six aggressiveness types based on Capela Dias et al. (2013). Maximum emergency deceleration was uniformly set to $-9.0\,\mathrm{m/s^2}$, as they did not specify this parameter.

Type	**Max. accel.**	**Max. decel.**	**Max. emerg. decel.**	**Min. gap**	**Min. headway**
Aggressive young	$3.1\,\mathrm{m/s^2}$	$-5.5\,\mathrm{m/s^2}$	$-9.0\,\mathrm{m/s^2}$	1.2 m	1.0 s
Courteous young	$2.5\,\mathrm{m/s^2}$	$-4.5\,\mathrm{m/s^2}$	$-9.0\,\mathrm{m/s^2}$	2.5 m	1.0 s
Aggressive middle-aged	$2.9\,\mathrm{m/s^2}$	$-5.0\,\mathrm{m/s^2}$	$-9.0\,\mathrm{m/s^2}$	2.0 m	1.3 s
Courteous middle-aged	$2.4\,\mathrm{m/s^2}$	$-4.1\,\mathrm{m/s^2}$	$-9.0\,\mathrm{m/s^2}$	2.5 m	1.5 s
Aggressive old	$2.6\,\mathrm{m/s^2}$	$-4.5\,\mathrm{m/s^2}$	$-9.0\,\mathrm{m/s^2}$	2.0 m	1.7 s
Courteous old	$2.3\,\mathrm{m/s^2}$	$-3.8\,\mathrm{m/s^2}$	$-9.0\,\mathrm{m/s^2}$	2.5 m	1.9 s

Several common parameters have varying effects on different car-following models. CityFlow assumes that vehicles have usual and maximum accelerations and decelerations; SUMO assumes that vehicles have a maximum possible acceleration and deceleration (which are used as the usual values), and a maximum emergency deceleration. The latter formulation was used here, although this may have resulted in more aggressive behavior than if lower usual accelerations and decelerations were used. Additionally, both simulators model vehicles as having minimum desired following distances in terms of space (i.e., the minimum gap) and time (i.e., the minimum headway). [24] co-varied these parameters to model the effect of driver aggressiveness on travel time. This work adopts their taxonomy of aggressiveness types, but excludes gender effects due to conflicting conclusions regarding their significance in the literature [139, 142]. This gave six parameter settings (1).

The default car-following models in CityFlow and SUMO are both modified from the collision avoidance model of [70]. For free speed, CityFlow's implementation uses the maximum speed, while SUMO's implementation modulates this by the visible lookahead distance. For stopping speed, both simulators solve somewhat different quadratic equations to determine the deceleration needed to stop within a fixed distance. For following speed, a target distance is maintained, which is computed by the desired minimum headway as well as the speed and maximum deceleration of the lead vehicle. In this work, SUMO's variant of the Krauß model was added to CityFlow. SUMO also implements other car-following models, which were also re-implemented in CityFlow to introduce variation in driver behavior:

- The collision avoidance/action point model of [150], which probabilistically combines Krauß's model with the action point model of [144].
- The collision avoidance model of [161], which is a behavioral model that varies

between free, approaching, following, and emergency modes based on the gap to the lead vehicle.

- The adaptive cruise control (ACC) model of [96], which determines acceleration using a "speed control" method if the gap to the lead vehicle is large and using a "gap control" method if the gap is small.

3.3.2 Lane-Changing Models

In both CityFlow and SUMO, lane changes are initiated by signaling the lead and lag vehicles on the destination lane, and the vehicle is instantly moved to the other lane upon completion of the lane change. (SUMO also supports a sublane model that explicitly models lateral movement, which was excluded as a control.) However, implementation details again differ between these simulators. Vehicles in CityFlow only perform lane changes to follow their routes, and lane-changing is based on the explicit insertion of a vehicle copy (the "shadow vehicle") on the target lane that the lag vehicle will follow. Vehicles in SUMO follow a hierarchy of motivations, which includes strategic lane changes to follow routes and tactical lane changes for overtaking.

One of SUMO's parameters, the gap tolerance factor, was introduced to CityFlow in this work. When vehicles are changing lanes, the minimum gap on the destination lane required for a vehicle to initiate a lane change is given by the necessary gap for collision avoidance divided by this constant factor. A factor of 1.0 represents the default behavior. The virtual experiments in this work varied this factor between 0.5, 0.82, 1.0, 1.18, and 1.5, representing decreases/increases of 18% and 50% in tolerance. This is based on [134], who observed the mean gaps for forced, cooperative, and free lane change maneuvers to be 45 ft, 53 ft, and 109 ft in a video dataset.

3.4 Experimental Setup

To address the research questions, two virtual experiments were designed. Both address RQ1 and RQ2 by assessing CityFlow and SUMO's distributional equivalence under different settings of driver behavior. The two experiments also address separate aspects of RQ3: Experiment 1 assesses distributional equivalence while varying the simulation scale by traffic demand, and Experiment 2 varies it by road network size. This was accomplished using road networks drawn from the benchmark dataset of [10], as described in the following subsections. All of these networks were initially defined using SUMO syntax; the SUMO-to-CityFlow network converter of [177] was used to generate their CityFlow counterparts, and created a flow converter to map between vehicular flows in the two simulators.

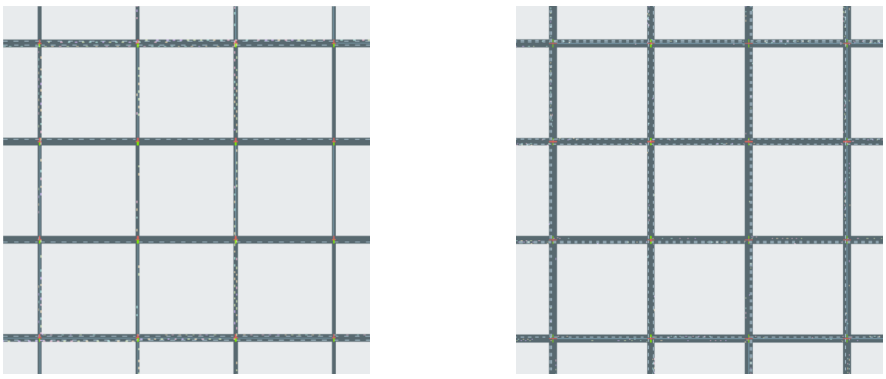
Four independent variables were considered in both virtual experiments: the car-following model (Section 3.3.1, 5 levels), the car-following aggressiveness parameters (Section 3.3.1, 6 levels), the lane-changing gap tolerance (Section 3.3.2, 5 levels), and the road network (2 levels). For the car-following model, the default and SUMO-based implementations of the Krauß model in CityFlow were compared to the SUMO Krauß model, while the other models were compared to their respective SUMO counterparts.

The fundamental lane-changing models for the two simulators were held constant, as was the traffic signal program — a simple fixed-time program from the original dataset.

Eight instantiations of two types of measures were used to assess distributional equivalence. First, the root mean squared error (RMSE) quantifies the point-to-point difference in individual outcome measures. This was computed as the mean RMSE of the total travel time and waiting time (defined as time that vehicles spend stopped with a speed $< 0.1^{\text{m}}/_{\text{s}}$) over all vehicles and timesteps; as the mean RMSE of per-lane total vehicle counts and queued vehicle counts over all lanes and timesteps; and as the mean RMSE of the speed and acceleration over all individual vehicles and timesteps. Second, the Kullback-Liebler (KL) divergence measures the difference in the distribution of outcomes over the entire road network. This was computed for the distributions of vehicle counts and queued vehicle counts as the mean over all timesteps.

3.4.1 Experiment 1: Traffic Demand

For this experiment, the road networks arterial4x4 [89] and grid4x4 [26] (1) from [10]'s RESCO benchmark were used. Both are synthetic grid networks with similar topologies, but they vary in the level of congestion. grid4x4 consists of six-lane roads with a uniformly distributed demand of 1,473 vehicles. arterial4x4 consists of major (four-lane) and minor (two-lane) roads, and a demand of 2,484 vehicles that alternates between major and minor roads. arterial4x4's traffic pattern empirically leads to congestion and degraded RL performance [10].



(a) **arterial4x4** (b) **grid4x4**

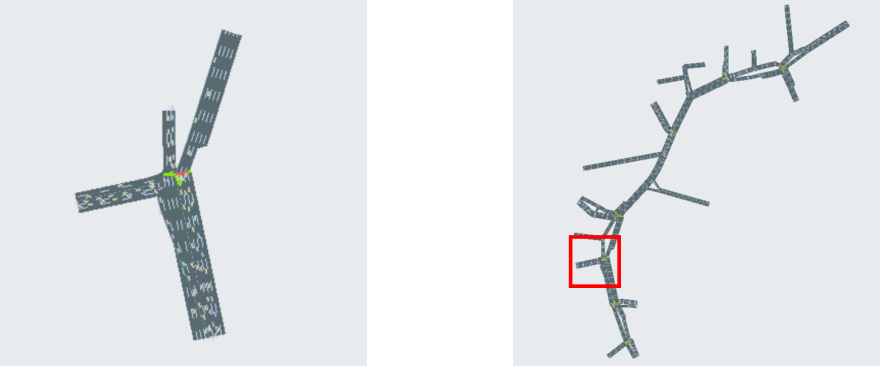
Figure 1: Screenshots in CityFlow of the arterial4x4 and grid4x4 road networks.

The following power analysis uses these variable names: C for car-following models, A for car-following aggressiveness, L for lane-changing gap tolerance, and R for road network. The second-order linear multiple regression included 5 (C) + 6 (A) + 1 (L) + 1 (L^2) + 2 (R) + 10 ($C \cdot R$) + 12 ($A \cdot R$) + 2 ($L \cdot R$) + 30 ($C \cdot A$) + 5 ($C \cdot L$) + 6 ($A \cdot L$) = 80 variables. Using G*Power 3.1.9.7's power calculation for ordinary linear multiple regression with a fixed model and R^2 increase, a small effect size of 0.02, and $\alpha = \beta = 0.95$, the total necessary sample size was computed as 2,646. Divided by the

number of cells, $5 \cdot 6 \cdot 5 \cdot 2 = 300$, the number of replications per cell was computed as $\lceil \frac{2,646}{300} \rceil = 9$. All replications were executed using Python 3.9.16, SUMO 1.12.0, and a modified version of CityFlow 0.1 on a shared server with four cores, two 4.2GHz Intel i7-7700K processors per core, and 62 GiB of RAM.

3.4.2 Experiment 2: Network Scale

For this experiment, the road networks ingolstadt1 and ingolstadt7 (2) from [10]'s RESCO benchmark were used. Both are subsets of the Ingolstadt road network that was simulated by [85]. They respectively contain 1 and 7 signalized intersections, representing a single intersection and a larger arterial road; ingolstadt7 is a superset of ingolstadt1. The total demands of the two road networks are respectively 1,716 and 3,031 vehicles.



(a) **ingolstadt1** (b) **ingolstadt7**

Figure 2: Screenshots in CityFlow of the ingolstadt1 and ingolstadt7 road networks.

Based on the analysis in Section 3.4.1, the number of replications per cell was computed as $\lceil \frac{2,646}{300} \rceil = 9$. All replications were executed using Python 3.9.16, SUMO 1.12.0, and a modified version of CityFlow 0.1 on a shared server with four cores, two 4.2GHz Intel i7-7700K processors per core, and 62 GiB of RAM.

3.5 Experimental Results

One-sample t-tests indicated that all of the RMSE and KL divergence measures were significantly different from 0, with a p-value $\ll 0.001$ for all cells in Experiments 1 and 2. This suggests a lack of distributional equivalence between CityFlow and SUMO. The following subsections explore the results of second-order linear multiple regression for each of the measures. Coefficients involving the difference between the SUMO and CityFlow Krauß model implementations were generally not significant.

3.5.1 Experiment 1: Traffic Demand

For total time and waiting time RMSEs, the road network, car-following model, and aggressiveness generally had significant effects, as did various pairwise interactions between them. The uncongested grid4x4 network had significantly lower RMSEs (coefficients: $-890.61/-1035.46$) than the congested arterial4x4 network (intercepts: 1718.61/1815.46), suggesting that congestion worsened the distributional equivalence of these measures. In arterial4x4, the Wagner (coefficients relative to SUMO Krauß: 546.8/466.65) and Wiedemann models (coefficients: 100.96/208.99) had significantly higher RMSEs; the differences were smaller for grid4x4 (coefficients relative to SUMO Krauß: 49.5/33.75 for Wagner; 48.91/34.85 for Wiedemann).

For total and queued vehicle count RMSEs and KL divergences, the road network, car-following model, aggressiveness, and gap tolerance generally had significant effects, as did various pairwise interactions between them. The RMSEs and KL divergences showed distinct patterns: the RMSEs were much lower for grid4x4 than arterial4x4 (coefficients: $-5.15/-4.93$), but the KL divergences had less variation (coefficients: $-0.621/-0.013$). Yet, the KL divergences had low enough standard deviations that the road network's effects remained significant. High aggressiveness in arterial4x4 generally yielded higher RMSEs (coefficients for aggressive young relative to aggressive middle-aged: 2.16/2.46) and KL divergences (coefficients: 0.228/0.139). While the same was true for the RMSEs in grid4x4, its KL divergences were lower for more aggressive settings (coefficients: $-0.204/-0.122$). Despite higher measures for time, the Wagner model led to lower measures for total vehicle count (coefficients relative to SUMO Krauß: $-1.16/-0.193$).

For vehicle speed and acceleration RMSEs, the road network, car-following model, and aggressiveness generally had significant effects, as did various pairwise interactions between them. Greater equivalence in vehicle distributions did not always correspond to more similar vehicle-level measures. Both measures increased for grid4x4 (coefficients: 2.62/0.418) even though the other measures were lower on average for this road network. Also, unlike its vehicle count measures but like its time measures, the Wagner model had significantly higher speed and acceleration RMSEs (coefficients relative to SUMO Krauß: 2.71/1.26).

3.5.2 Experiment 2: Network Scale

For total time and waiting time RMSEs, the aggressiveness and its interactions generally had significant effects, along with the Wiedemann model and its interactions. The best-fitting model for total time did not include a road network-gap tolerance interaction, whereas the model for waiting time did. The smaller ingolstadt1 network had lower but more variable RMSEs (intercepts: 899.29/1170.3), while the larger ingolstadt7 network had significantly higher but more uniform RMSEs (coefficients: 1264.59/1046.07). For ingolstadt1, aggressive parameter settings led to higher RMSEs (coefficients of aggressive young relative to aggressive middle-aged: 408.36/568.65). Also, the Wiedemann model had significantly higher RMSEs than other car-following models in ingolstadt1 (coefficients relative to SUMO Krauß: 192.29/1917.31), but this effect was reversed for ingolstadt7 (coefficients: $-349.13/-464.13$).

For total and queued vehicle count RMSEs and KL divergences, the road network and aggressiveness generally had significant effects, as did various pairwise interactions between them and with the gap tolerance. Unlike Experiment 1, the RMSEs measures were lower in ingolstadt7 than in ingolstadt1 (coefficients: $-2/-1.67$), but the KL divergence measures were higher (coefficients: 0.461/2.48). More aggressive parameter settings again led to significant increases in the RMSEs and KL divergences, with a larger increase in RMSEs (coefficients of aggressive young relative to aggressive middle-aged: 1.35/2.29) than in KL divergences (coefficients: 0.595/0.717). However, the increase in both measures was smaller in ingolstadt7 (coefficients: 0.764/1.507 for RMSE, 0.283/0.413 for KL divergences). Both the Wagner and ACC models had RMSEs that significantly increased with gap tolerance (coefficients relative to SUMO Krauß per unit of gap tolerance: 0.151/0.056 for Wagner, 0.165/0.092 for ACC), but KL divergences that significantly decreased with it (coefficients: $-0.285/-0.511$ for Wagner, $-0.354/-0.642$ for ACC).

For vehicle speed and acceleration RMSEs, the Wagner and Wiedemann models along with the aggressiveness had significant effects, as did various pairwise interactions of the road network with the car-following models and aggressiveness. Again, the road network had significant effects on both speed and acceleration RMSEs, with these measures being higher for ingolstadt7 (coefficients: 5.51/0.173). The Wagner car-following model had significantly higher RMSEs (coefficients relative to SUMO Krauß: 0.363/0.461), whereas the Wiedemann model had significantly lower RMSEs (coefficients: $-1.94/-0.957$).

3.5.3 Parameter Validity

Validation was conducted by comparing the parameter settings for car-following and lane-changing models used in the virtual experiments to prior driving simulator and real-world studies. Overall, the dependence of these parameters on external factors such as traffic density and speed suggests that the settings used in traffic simulators should be calibrated to specific road networks and conditions. However, the settings used in this work remain reasonable considering the variation reported in the literature.

For acceleration and deceleration, the settings used in the virtual experiments were based on [24], with smaller values for older and less aggressive drivers; however, they considered gender effects to be negligible. Similar values have been reported in prior work [106, 39, 67]. However, among driving simulator studies, [67] demonstrated an age effect opposite to that assumed by Capela Dias et al. Among real-world studies, [99] found a dependence of the 95th percentile of braking decelerations on speed, and [106] reported a significant interaction between age and gender.

For minimum gap and headway time, the settings used in the virtual experiments also followed [24], with larger values for older and less aggressive drivers. Similar values have again been reported in prior work [95, 149]. In particular, real-world studies for which reported values are influenced by a similar age effect include [31, 139], and [99]. [139] and [123] reported larger minimum gaps and smaller minimum headways, but their measurements were made for higher speeds (over 50 $\mathrm{km/h}$ and 100 $\mathrm{km/h}$).

For lane-changing gap tolerance, this work used gap widths in empirical lane-changing behavior to approximately quantify variation, specifically the mean gaps for

forced, cooperative, and free lane changes from [134]'s real-world study. The settings in this work are consistent with standard deviations in lead and lag gaps as reported in prior studies [101, 11]. [4]'s driving simulator study identified significant factors that impact gap tolerance: relative to the average male middle-aged driver, gaps are smaller for younger drivers, larger for female drivers, and smaller as speed increases. Likewise, [58]'s real-world study reported that the mean and standard deviation of lag gaps depended on congestion. Future work could use these factors to create a taxonomy of lane-changing behavior similar to [24].

3.6 Discussion

In this work, virtual experiments compared the low-level simulation outcomes of two traffic simulators, CityFlow and SUMO. To capture the effects of modelling real-world heterogeneity, various parameters of driver behavior and road network scale were varied. The results indicate a lack of distributional equivalence between the simulators, with certain parameter settings worsening distributional equivalence.

However, as noted in Section 3.2, this work is insufficient to provide a complete characterization of what the critical differences between these simulators are. Many aspects of CityFlow and SUMO that were not controlled — simulation control-flow, other aspects of driver behavior, and the effects of traffic signals — could all have contributed to the observed discrepancies. Thus, future work should perform more comprehensive, controlled evaluations of these two simulators.

Regardless, researchers in RL for transportation must not take traffic simulators for granted as a deus ex machina for training, and must recognize that they may not be interchangeable. Which simulator, then, should be chosen? This work does not aim to answer this question, but some observations can be made:

- **SUMO** provides a detailed simulation that models real-world heterogeneity, and captures additional aspects of traffic management and driver behavior
- **CityFlow** provides an efficient simulation that abstracts out and homogenizes various details, reducing the number of parameters that need to be tuned

Therefore, as with many other problems in simulation, the core trade-off between these two simulators (and others) involves veridicality and efficiency. There is no one best simulator; researchers must decide whether using a coarser abstraction of the environment is acceptable in exchange for faster training.

But how exactly should researchers make this decision? Crucially, RL-based ITSs may not necessarily perform better when they are trained with more granular simulators. Prior work has shown that both introducing unnecessary complexity [184] and removing needed complexity [47] in the observation space may harm performance. Researchers should compare training results from different simulators and use them to design RL formulations in a principled way. As a first step, [50] report training results on both SUMO and CityFlow, but they do not use the same road networks for both simulators. One strategy may be to train a baseline using an efficient simulator (e.g., CityFlow), and then to finetune it by further training with a veridical one (e.g., SUMO).

Ultimately, the true goal is to ensure that RL-based ITSs can perform well under real-world traffic conditions, which requires that their training and validation environments are close to reality. As previously noted, distributional equivalence between simulators is a proxy measure of this goal. Unfortunately, a chicken-and-egg problem exists in that traffic simulators are intended to replace real-world data collection, yet cannot be validated without it. For now, simulations should still be developed in collaboration with stakeholders to ensure that they meet acceptable standards of fidelity. However, the future holds promise for traffic simulations that are both veridical and efficient: the increasing prevalence of connected vehicles [94] means that collection of granular real-world data for validation may be within reach.

4 Ongoing and Future Work

We built the simulation environment for traffic in SUMO which includes 36 intersections in the center area of Strongsville, OH, and refined it based on feedback from our collaborators at Econolite and Path Master. This environment is built in a way that matches the real-world data of vehicle counts every minute in each lane at each of the intersections. In addition, we trained, evaluated, and compared the performance of three different traffic signaling schemes: (1) the default one currently used by Strongsville; (2) the scheme trained with the state-of-the-art MARL algorithm [26]; (3) a heuristic scheme that greedily chooses a phase. The results show that MARL can indeed lead to meaningful improvements over the default scheme. While our results are promising, there is still significant room for further improvement. We are currently focusing on improving the interpretability of MARL-generated policies and have proposed an initial algorithm that adapts the recently proposed MAVIPER algorithm [97] to generate decision-tree policies for TSC.

5 Individuals Involved

Table 2: Individuals Involved

Name	Role	ORCID
Fei Fang	PI	0000-0003-2256-8329
Norman Sadeh	Co-PI	0000-0003-4829-5533
Rex H.-G. Chen	Ph.D. Student	0000-0002-1620-0440
Kathleen Carley	Collaborator	0000-0002-6356-0238

6 Publications

1. The Real Deal: A Review of Challenges and Opportunities in Moving Reinforcement Learning-Based Traffic Signal Control Systems Towards Reality. Rex Chen, Fei Fang and Norman Sadeh. ATT'22: Workshop on Agents in Traffic

and Transportation, July 25, 2022, Vienna, Austria. [Arxiv Version: https://arxiv.org/pdf/2206.11996.pdf[Proceeding Version: https://ceur-ws.org/Vol-3173/2.pdf]

2. Purpose in the Machine: Do Traffic Simulators Produce Distributionally Equivalent Outcomes for Reinforcement Learning Applications? Rex Chen, Kathleen M. Carley, Fei Fang, Norman Sadeh. Proceedings of the 2023 Winter Simulation Conference.

References

[1] Monireh Abdoos and Ana L.C. Bazzan. Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. Expert Systems with Applications, 171:114580, 2021.

[2] Baher Abdulhai and Lina Kattan. Reinforcement learning: Introduction to theory and potential for transport applications. Canadian Journal of Civil Engineering, 30:981–991, 2003.

[3] Lucas N. Alegre, Ana L.C. Bazzan, and Bruno C. da Silva. Quantifying the impact of non-stationarity in reinforcement learning-based traffic signal control. PeerJ Computer Science, 7:e575, 2021.

[4] Yasir Ali, Zuduo Zheng, Md. Mazharul Haque, Mehmet Yildirimoglu, and Simon Washington. Understanding the discretionary lane-changing behaviour in the connected environment. Accident Analysis & Prevention, 137:105463, 2020.

[5] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using SUMO. In Proceedings of the 21st International Conference on Intelligent Transportation Systems, ITSC '18, pages 2575–2582, Piscataway, 2018. Institute of Electrical and Electronics Engineers.

[6] Christopher Amato. Decision-making under uncertainty in multi-agent and multi-robot systems: Planning and learning. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18, pages 5662–5666, 2018.

[7] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. The International Journal of Robotics Research, 39:3–20, 2020.

[8] Mohammad Aslani, Stefan Seipel, Mohammad Saadi Mesgari, and Marco Wiering. Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran. Advanced Engineering Informatics, 38:639–655, 2018.

[9] James Ault, Josiah P. Hanna, and Guni Sharon. Learning an interpretable traffic signal control policy. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20, pages 88–96, 2020.

[10] James Ault and Guni Sharon. Reinforcement learning benchmarks for traffic signal control. In Proceedings of the 35th Conference on Neural Information Processing Systems, Track on Datasets and Benchmarks, NeurIPS '21, pages 1–11. Neural Information Processing Systems, 2021.

[11] Esmaeil Balal, Ruey Long Cheu, Thompson Gyan-Sarkodie, and Jessica Miramontes. Analysis of discretionary lane changing parameters on freeways. International Journal of Transportation Science and Technology, 3(3):277–296, 2014.

[12] J. Barceló, E. Codina, J. Casas, J.L. Ferrer, and D. García. Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems. Journal of Intelligent and Robotic Systems, 41:173–203, 2004.

[13] Jaume Barceló. Models, traffic models, simulation, and traffic simulation. In Fundamentals of Traffic Simulation, pages 1–62. Springer, 2010.

[14] Nelson Baza-Solares, Ruben Velasquez-Martínez, Cristian Torres-Bohórquez, Yerly Martínez-Estupiñán, and Cristian Poliziani. Traffic simulation with open-source and commercial traffic microsimulators: A case study. Communications — Scientific Letters of the University of Zilina, 24(2):E49–E62, 2022.

[15] Ana L. C. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Autonomous Agents and Multi-Agent Systems, 18:342–375, 2009.

[16] Ana L. C. Bazzan and Franziska Klügl. A review on agent-based technology for traffic and transportation. The Knowledge Engineering Review, 29(3):375–403, 2013.

[17] Luca Bedogni, Marco Gramaglia, Andrea Vesco, Marco Fiore, Jérôme Härri, and Francesco Ferrero. The Bologna ringway dataset: Improving road network conversion in SUMO and validating urban mobility via navigation services. IEEE Transactions on Vehicular Technology, 64(12):5464–5476, 2015.

[18] Steven Bohez, Abbas Abdolmaleki, Michael Neunert, Jonas Buchli, Nicolas Heess, and Raia Hadsell. Value constrained model-free continuous control. arXiv preprint, 2019.

[19] James Bonneson, Michael Pratt, and Karl Zimmerman. Development of a traffic signal operations handbook. Technical Report FHWA/TX-09/0-5629-1, Texas A&M Transportation Institute, 2009.

[20] David Branston and Henk van Zuylen. Comparison of queue-length models at signalized intersections. Transportation Research, 12(1):47–53, 1978.

[21] Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. Science, 359(6374):418–424, 2017.

[22] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems, 5, 2022.

[23] Chen Cai, Bernhard Hengst, Getian Ye, Enyang Huang, Yang Wang, Carlos Aydos, and Glenn Geers. On the performance of adaptive traffic signal control. In Proceedings of the Second International Workshop on Computational Transportation Science, ICWTS '09, pages 37–42, 2009.

[24] José António Capela Dias, Penousal Machado, and Francisco Câmara Pereira. Simulating the impact of drivers' personality on city transit. In Proceedings of the 13th World Conference on Transport Research, WCTR '13, pages 1–13, Leeds, 2013. World Conference on Transport Research Society.

[25] Pitipong Chanloha, Jatuporn Chinrungrueng, Wipawee Usaha, and Chaodit Aswakul. Cell transmission model-based multiagent Q-learning for network-scale signal control with transit priority. The Computer Journal, 57(3):451–468, 2014.

[26] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI '20, pages 3414–3421, 2020.

[27] Hongyun Chen and Jian John Lu. Comparison of current practical adaptive traffic control systems. In Proceedings of the 10th International Conference of Chinese Transportation Professionals, ICCTP '10, pages 1611–1619, 2010.

[28] S.M. Chin, O. Franzese, D.L. Greene, H.L. Hwang, and R.C. Gibson. Temporary losses of highway capacity and impacts on performance: Phase 2. Technical Report ORNL/TM-2004/209, Oak Ridge National Laboratory, 2004.

[29] Tianshu Chu and Jie Wang. Traffic signal control by distributed reinforcement learning with min-sum communication. In Proceedings of the 2017 American Control Conference, ACC '17, pages 5095–5100, 2017.

[30] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems, 21:1086–1095, 2020.

[31] Delphine Delorme and Bongsob Song. Human driver model for SmartAHS. Technical Report UCB-ITS-PRR-2001-12, Institute of Transportation Studies, University of California, Berkeley, Berkeley, 2001.

[32] Konstantinos Dimitropoulos, Ioannis Hatzilygeroudis, and Konstantinos Chatzilygeroudis. A brief survey of Sim2Real methods for robot learning. In Proceedings of the 2022 International Conference on Robotics in Alpe-Adria Danube Region, RAAD '22, pages 133–140, 2022.

[33] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. Natural policy gradient primal-dual method for constrained Markov decision processes. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NeurIPS '20, pages 8378–8390, 2020.

[34] DOT. Manual on Uniform Traffic Signal Control Devices. US Department of Transportation, revision 2 edition, 2012.

[35] A. M. Tahsin Emtenan and Christopher M. Day. Impact of detector configuration on performance measurement and signal operations. Transportation Research Record, 2674(4):300–313, 2020.

[36] Myungeun Eom and Byung-In Kim. The traffic signal control problem for intersections: a review. European Transport Research Review, 12:50, 2020.

[37] Mohamed Essa and Tarek Sayed. Traffic conflict models to evaluate the safety of signalized intersections at the cycle level. Transportation Research Part C: Emerging Technologies, 89:289–302, 2018.

[38] Mohamed Essa and Tarek Sayed. Self-learning adaptive traffic signal control for real-time safety optimization. Accident Analysis & Prevention, 146:105713, 2020.

[39] Karim Fadhloun, Hesham Rakha, Amara Loulizi, and Abdessattar Abdelkef. Vehicle dynamics model for estimating typical vehicle accelerations. Transportation Research Record, 2491:61–71, 2015.

[40] Ryan Fries, Imran Inamdar, Mashrur Chowdhury, Kevin Taaffe, and Kaan Ozbay. Feasibility of traffic simulation for decision support in real-time regional traffic management. Transportation Research Record, 2035:169–176, 2007.

[41] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16:1437–1480, 2015.

[42] Deepeka Garg, Maria Chli, and George Vogiatzis. Deep reinforcement learning for autonomous traffic light control. In Proceedings of the 2018 3rd International Conference on Intelligent Transportation Engineering, ICITE '18, pages 214–218, 2018.

[43] Deepeka Garg, Maria Chli, and George Vogiatzis. Traffic3D: A rich 3D-traffic environment to train intelligent agents. In Proceedings of the 19th International Conference on Computational Science, ICCS '19, pages 749–755, New York, 2019. Springer.

[44] Deepeka Garg, Maria Chli, and George Vogiatzis. Fully-autonomous, vision-based traffic signal control: from simulation to reality. In Proceedings of the 21th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '22, pages 454–462, 2022.

[45] Hongwei Ge, Yumei Song, Chunguo Wu, Jiankang Ren, and Guozhen Tan. Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control. IEEE Access, 7:40797–40809, 2019.

[46] Wade Genders and Saiedeh Razavi. Using a deep reinforcement learning agent for traffic signal control. arXiv preprint, 2016.

[47] Wade Genders and Saiedeh Razavi. Evaluating reinforcement learning state representations for adaptive traffic signal control. Procedia Computer Science, 130:26–33, 2018.

[48] Douglas Gettman, Steven G. Shelby, Larry Head, Darcy M. Bullock, and Nils Soyke. Data-driven algorithms for real-time adaptive tuning of offsets in coordinated traffic signal systems. Transportation Research Record, 2035(1):1–9, 2007.

[49] David Gibson, Milton K. (Pete) Mills, and Doug Rekenthaler Jr. Staying in the loop: The search for improved reliability of traffic sensing systems through smart test instruments. Public Roads, 62(2), 1998.

[50] Harsh Goel, Yifeng Zhang, Mehul Damani, and Guillaume Sartoretti. Sociallight: Distributed cooperation learning towards network-wide traffic signal control. In Proceedings of the 22nd International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '23, pages 1–10, London, 2023. International Foundation for Autonomous Agents and Multiagent Systems.

[51] Yang Miang Goh and Peter E.D. Love. Methodological application of system dynamics for evaluating traffic safety policy. Safety Science, 50(7):1594–1605, 2012.

[52] Yaobang Gong, Mohamed Abdel-Aty, Jinghui Yuan, and Qing Cai. Multi-objective reinforcement learning approach for improving safety at intersections with adaptive traffic signal control. Accident Analysis & Prevention, 144:105655, 2020.

[53] Robert L. Gordon and Warren Tighe. Traffic Control Systems Handbook. Federal Highway Administration, 2005.

[54] Martin Gregurić, Miroslav Vujić, Charalampos Alexopoulos, and Mladen Miletić. Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data. Applied Sciences, 10(11):4011, 2020.

[55] Ge Guo and Yunpeng Wang. An integrated MPC and deep reinforcement learning approach to trams-priority active signal control. Control Engineering Practice, 110:104758, 2021.

[56] Mengyu Guo, Pin Wang, Ching-Yao Chan, and Sid Askary. A reinforcement learning approach for intelligent traffic signal control at urban intersections. In Proceedings of the 2019 International Conference on Intelligent Transportation Systems, ITSC '19, pages 4242–4247, 2019.

[57] Ammar Haydari and Yasin Yilmaz. Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 23:11–32, 2022.

[58] Corey Hill, Lily Elefteriadou, and Alexandra Kondyli. Exploratory analysis of lane changing on freeways based on driver behavior. Journal of Transportation Engineering, 141(4):1–11, 2015.

[59] Zhitong Huang, Ed Leslie, and Animesh Balse. Infrastructure connectivity certification test procedures for infrastructure-based connected automated vehicle components: Test procedures, signal phase and timing — NTCIP 1202 v03. Technical Report FHWA-JPO-20-802, Leidos, 2019.

[60] S M A Bin Al Islam, Mehrdad Tajalli, Rasool Mohebifard, and Ali Hajbabaie. Effects of connectivity and traffic observability on an adaptive traffic signal control system. Transportation Research Record, 2675(10):800–814, 2021.

[61] Vindula Jayawardana, Anna Landler, and Cathy Wu. Mixed autonomous supervision in traffic signal control. In Proceedings of the 2021 International Conference on Intelligent Transportation Systems, ITSC '21, pages 1767–1773, 2021.

[62] Qize Jiang, Minhao Qin, Shengmin Shi, Weiwei Sun, and Baihua Zheng. Multi-agent reinforcement learning for traffic signal control through universal communication method. arXiv preprint, 2022.

[63] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 2021.

[64] Jelle R. Kok and Nikos Vlassis. Using the max-plus algorithm for multiagent decision making in coordination graphs. In Proceedings of the Fourth Robot Soccer World Cup, RoboCup '05, pages 1–12, 2005.

[65] Behrad Koohy, Sebastian Stein, Enrico Gerding, and Ghaithaa Manla. Reward function design in multi-agent reinforcement learning for traffic signal control. In Proceedings of the 12th International Workshop on Agents in Traffic and Transportation, ATT '22, pages 1–13, Vienna, 2022. International Joint Conference on Artificial Intelligence.

[66] Peter Koonce, Lee Rodegerdts, Kevin Lee, Shaun Quayle, Scott Beaird, Cade Braud, Jim Bonneson, Phil Tarnoff, and Tom Urbanik. Traffic Signal Timing Manual. Federal Highway Administration, 2008.

[67] Moritz Körber, Christian Gold, David Lechner, and Klaus Bengler. The influence of age on the take-over of vehicle control in highly automated driving. Transportation Research Part F: Traffic Psychology and Behaviour, 39:19–32, 2016.

[68] Sirisha Kothuri, Andrew Kading, Edward Smaglik, and Christopher Sobie. Improving walkability through control strategies at signalized intersections. Technical Report NITC-RR-782, National Institute for Transportation and Communities, 2017.

[69] Daniel Krajzewicz, Georg Hertkorn, C. Rössel, and Peter Wagner. SUMO (Simulation of Urban MObility) - an open-source traffic simulation. In Proceedings of the 4th Middle East Symposium on Simulation and Modelling, MESM '02, pages 183–187, 2002.

[70] Stefan Krauß. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. PhD thesis, German Aerospace Center, Cologne, 1998.

[71] Neetesh Kumar, Syed Shameerur Rahman, and Navin Dhakad. An integrated MPC and deep reinforcement learning approach to trams-priority active signal control. IEEE Transactions on Intelligent Transportation Systems, 22(8):4919–4928, 2021.

[72] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. In Proceedings of the 38th International Conference on Machine Learning, ICML '21, pages 6131–6141, 2021.

[73] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. WeBuildAI: Participatory framework for algorithmic governance. Proceedings of the ACM on Human-Computer Interaction, 3:1–35, 2019.

[74] Catur Yudo Leksono and Tina Andriyana. Roundabout microsimulation using sumo: A case study in idrottsparken roundabout, norrköping, sweden. Master's thesis, Linköping University, Linköping, 2012.

[75] Congcong Li, Fei Yan, Yiduo Zhou, Jia Wu, and Xiaomin Wang. A regional traffic signal control strategy with deep reinforcement learning. In Proceedings of the 37th Chinese Control Conference, CCC '18, pages 7690—7695, 2018.

[76] Wangzhi Li, Mobin Zhao, Yongjie Fu, Kangrui Ruan, and Xuan Di. CVLight: Decentralized learning for adaptive traffic signal control with connected vehicles. arXiv preprint, 2021.

[77] Yongfu Li and Dihua Sun. Microscopic car-following model for the traffic flow: the state of the art. Journal of Control Theory and Applications, 10:133–143, 2012.

[78] Yuxi Li. Deep reinforcement learning. arXiv preprint, 2018.

[79] Zhenning Li, Hao Yu, Guohui Zhang, Shangjia Dong, and Cheng-Zhong Xu. Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. Transportation Research Part C: Emerging Technologies, 125:103059, 2021.

[80] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. A deep reinforcement learning network for traffic light cycle control. IEEE Transactions on Vehicular Technology, 68(2):1243–1253, 2019.

[81] Lyuchao Liao, Jierui Liu, Xinke Wu, Fumin Zou, Jengshyang Pan, Qi Sun, Shengbo Eben Li, and Maolin Zhang. Time difference penalized traffic signal timing by LSTM Q-network to balance safety and capacity at intersections. IEEE Access, 8:80086–80096, 2020.

[82] Weirong Liu, Gaorong Qin, Yun He, and Fei Jiang. Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering. IEEE Transactions on Vehicular Technology, 66(10):8667–8681, 2017.

[83] Ying Liu, Lei Liu, and Wei-Peng Chen. Intelligent traffic light control using distributed multi-agent Q learning. In Proceedings of the 2017 International Conference on Intelligent Transportation Systems, ITSC '17, pages 1–8, 2017.

[84] Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Zhiwei Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In Proceedings of the 39th International Conference on Machine Learning, ICML '22, pages 1–9, 2022.

[85] Silas C. Lobo, Stefan Neumeier, Evelio M. G. Fernandez, and Christian Facchi. InTAS — the Ingolstadt traffic scenario for SUMO. In Proceedings of the 2020 SUMO User Conference, SUMO '20, pages 1–20, Cologne, 2020. German Aerospace Center.

[86] Dominique Lord and James A. Bonneson. Role and application of accident modification factors within highway design process. Transportation Research Record, 1961(1):65–73, 2006.

[87] Felipe Luyanda, Douglas Gettman, Larry Head, Steven Shelby, Darcy Bullock, and Pitu Mirchandani. ACS-Lite algorithmic architecture: Applying adaptive control system technology to closed-loop traffic signal control systems. Transportation Research Record, 1856(1):175–184, 2003.

[88] Jiaqi Ma, Michael D. Fontaine, Fang Zhou, and Jia Hu. Estimation of crash modification factors for an adaptive traffic-signal control system. Journal of Transportation Engineering, 142(12):04016061, 2016.

[89] Jinming Ma and Feng Wu. Feudal multi-agent deep reinforcement learning for traffic signal control. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20, pages 816–824, 2020.

[90] Michal Maciejewski. A comparison of microscopic traffic flow simulation systems for an urban area. Transport Problems, 5(4):27–38, 2010.

[91] Aravindh Mahendran, Stephen Smith, Martial Hebert, and Xiao-Feng Xie. Bus detection for adaptive traffic signal control. Technical report, Carnegie Mellon University, 2014.

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

[93] Gustav Markkula, Ola Benderius, Krister Wolff, and Mattias Wahde. A review of near-collision driver behavior models. Human Factors, 54(6):1117–1143, 2012.

[94] Jijo Mathew, Jairaj Desai, Rahul Suryakant Sakhare, Woosung Kim, Howell Li, and Darcy M. Bullock. Big data applications for managing roadways. Institute of Transportation Engineers Journal, 91(2):28–35, 2021.

[95] Paul G Michael, Frank C Leeming, and William O Dwyer. Headway on urban streets: observational data and an intervention to decrease tailgating. Transportation Research Part F: Traffic Psychology and Behaviour, 3:55–64, 2000.

[96] Vicente Milanés and Steven E. Shladover. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. Transportation Research Part C: Emerging Technologies, 48:285–300, 2014.

[97] Stephanie Milani, Zhicheng Zhang, Nicholay Topin, Zheyuan Ryan Shi, Charles Kamhoua, Evangelos E Papalexakis, and Fei Fang. Maviper: Learning decision tree policies for interpretable multi-agent reinforcement learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 251–266. Springer, 2022.

[98] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. Nature, 518:529–533, 2015.

[99] Seungwuk Moon and Kyongsu Yi. Human driving data-based design of a vehicle adaptive cruise control algorithm. Vehicle System Dynamics, 46(8):661–690, 2008.

[100] Sara Moridpour, Majid Sarvi, and Geoff Rose. Lane changing models: a critical review. Transportation Letters, 2(3):157–173, 2010.

[101] Sara Moridpour, Majid Sarvi, Geoff Rose, and Euan Ramsay. Variables influencing lane changing behaviour of heavy vehicles. In Proceedings of the 31st Australasian Transport Research Forum, ATRF '08, pages 1–15, Canberra, 2008. Australasian Transport Research Forum.

[102] Jean-Baptiste Mouret and Konstantinos Chatzilygeroudis. 20 years of reality gap: a few thoughts about simulators in evolutionary robotics. In Proceedings of the 2017 Genetic and Evolutionary Computation Conference Companion, GECCO '17, pages 1121–1124, 2017.

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

[104] Arthur Müller, Vishal Rangras, Georg Schnittker, Michael Waldmann, Maxim Friesen, Tobias Ferfers, Lukas Schreckenberg, Florian Hufen, Jürgen Jasperneite, and Marco Wiering. Towards real-world deployment of reinforcement learning for traffic signal control. In Proceedings of the 20th IEEE International Conference on Machine Learning and Applications, ICMLA '21, pages 507–514, 2021.

[105] Nadia Nahar, Shurui Zhou, Grace Lewis, and Christian Kästner. Collaboration challenges in building ML-enabled systems: Communication, documentation, engineering, and process. In Proceedings of the 44th International Conference on Software Engineering, ICSE '22, pages 1–22, 2022.

[106] Satoru Nakagawa, Dean Kriellaars, Christine Blais, Jeannette Montufar, and Michelle M. Porter. Speed and acceleration patterns of younger and older drivers. Technical report, University of Manitoba, Winnipeg, 2006.

[107] Mohammadreza Nazari, Afshin Oroojlooy, Martin Takáč, and Lawrence V. Snyder. Reinforcement learning for solving the vehicle routing problem. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS '18, pages 9861–9871, 2018.

[108] Tomoki Nishi, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura. Traffic signal control based on reinforcement learning with graph convolutional neural nets. In Proceedings of the 2018 International Conference on Intelligent Transportation Systems, ITSC '18, pages 877–883, 2018.

[109] Mohammad Noaeen, Atharva Naik, Liana Goodman, Jared Crebo, Taimoor Abrar, Zahra Shakeri Hossein Abad, Ana L.C. Bazzan, and Behrouz Far. Reinforcement learning in urban network traffic signal control: A systematic literature review. Expert Systems with Applications, 199:116830, 2022.

[110] Venktesh Pandey, Evana Wang, and Stephen D. Boyles. Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. Transportation Research Part C: Emerging Technologies, 119:102715, 2020.

[111] Andrew Perrault, Fei Fang, Arunesh Sinha, and Milind Tambe. AI for social impact: Learning and planning in the data-to-deployment pipeline. arXiv preprint, 2019.

[112] J. Peters, P. O'Brien, and J. Pachman. Memorandum: Farmington Road adaptive traffic control benefits analysis. Technical report, DKS Associates, 2011.

[113] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché-Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 Reproducibility Program). Journal of Machine Learning Research, 22:1–20, 2021.

[114] Zhiwei (Tony) Qin, Xiaocheng Tang, Yan Jiao, Fan Zhang, Zhe Xu, Hongtu Zhu, and Jieping Ye. Ride-hailing order dispatching at didi via reinforcement learning. INFORMS Journal on Applied Analytics, 50(5):272–286, 2020.

[115] Faizan Rasheed, Kok-Lim Alvin Yau, Rafidah Md. Noor, Celimuge Wu, and Yeh-Ching Low. Deep reinforcement learning for traffic signal control: A review. IEEE Access, 8:208016–208044, 2020.

[116] Avery Rhodes, Darcy M. Bullock, James R. Sturdevant, and Zachary Thomas Clark. Evaluation of stop bar video detection accuracy at signalized intersections. Technical Report FHWA/IN/JTRP-2005/28, Joint Transportation Research Program, Indiana Department of Transportation and Purdue University, 2005.

[117] Stefano Giovanni Rizzo, Giovanna Vantini, and Sanjay Chawla. Reinforcement learning with explainability for traffic signal control. In Proceedings of the 2019 International Conference on Intelligent Transportation Systems, ITSC '19, pages 3567–3572, 2019.

[118] Filipe Rodrigues and Carlos Lima Azevedo. Towards robust deep reinforcement learning for traffic signal control: Demand surges, incidents and sensor failures. In Proceedings of the 2019 International Conference on Intelligent Transportation Systems, ITSC '19, pages 3559–3566, 2019.

[119] Aji Ronaldo and M. Taufiq Ismail. Comparison of the two micro-simulation software AIMSUN & SUMO for highway traffic modelling. Master's thesis, Linköping University, Linköping, 2012.

[120] Mohammad Saifuzzaman and Zuduo Zheng. Incorporating human-factors in car-following models: A review of recent developments and research needs. Transportation Research Part C: Emerging Technologies, 48:379–403, 2014.

[121] David Schrank, Luke Albert, Bill Eisele, and Tim Lomax. 2021 Urban Mobility Report. Technical report, Texas A&M Transportation Institute, 2021.

[122] Soheil Mohamad Alizadeh Shabestray and Baher Abdulhai. Multimodal iNtelligent Deep (MiND) traffic signal controller. In Proceedings of the 2019 International Conference on Intelligent Transportation Systems, ITSC '19, pages 4532–4539, 2019.

[123] Qiangqiang Shangguan, Xuekun Wang, Shuo Liu, and Junhua Wang. Car following behavior under foggy conditions with different road alignments — a driving simulator based study. In Proceedings of the 5th International Conference on Transportation Information and Safety, ICTS '19, pages 127–135, Piscataway, 2019. Institute of Electrical and Electronics Engineers.

[124] Anuj Sharma, Edward Smaglik, Sirisha Kothuri, Oliver Smith, Peter Koonce, and Tingting Huang. Leading pedestrian intervals: Treating the decision to implement as a marginal benefit–cost problem. Transportation Research Record, 2620(1):96–104, 2017.

[125] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. Nature, 529:484–489, 2016.

[126] Courtney Slavin, Wei Feng, Miguel Figliozzi, and Peter Koonce. Statistical study of the impact of adaptive traffic signal control on traffic and transit performance. Transportation Research Record, 2356(1):117–126, 2016.

[127] Stephen Smith. Surtrac for the people: Upgrading the Surtrac Pittsburgh deployment to incorporate pedestrian friendly extensions and remote monitoring advances. Technical Report 01730614, Mobility21, 2020.

[128] Stephen Smith, Gregory Barlow, Xiao-Feng Xie, and Zack Rubinstein. Smart urban signal networks: Initial application of the SURTRAC adaptive traffic signal control system. In Proceedings of the 23rd International Conference on Automated Planning and Scheduling, ICAPS '13, pages 434–442, 2013.

[129] Stephen Smith, Isaac Isukapati, Eli Bronstein, and Conor Igoe. Integrating transit signal priority with adaptive signal control in a connected vehicle environment: Phase 1 final report. Technical Report 01675986, Mobility21, 2018.

[130] Matthijs T. J. Spaan and Nikos Vlassis. A point-based pomdp algorithm for robot planning. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA '04, pages 2399–2404, 2004.

[131] Haoran Su, Kejian Shi, Joseph Chow, and Li Jin. Dynamic queue-jump lane for emergency vehicles under partially connected settings: A multi-agent deep reinforcement learning approach. arXiv preprint, 2021.

[132] Haoran Su, Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. EMVLight: A decentralized reinforcement learning framework for efficient passage of emergency vehicles. In Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI '22, pages 1–11, 2022.

[133] Carlos Sun, Henry Brown, Praveen Edara, Boris Carlos, and Kyoungmin Nam. Calibration of the *Highway Safety Manual* for Missouri. Technical Report 25-1121-0003-177, Mid-America Transportation Center, 2013.

[134] Daniel (Jian) Sun and Alexandra Kondyli. Modeling vehicle interactions during lane-changing behavior on arterial streets. Computer-Aided Civil and Infrastructure Engineering, 25:557–571, 2010.

[135] Dazhi Sun, Leslie Dodoo, Andres Rubio, Harsha Kalyan Penumala, Michael Pratt, and Srinivasa Sunkari. Synthesis study of texas signal control systems: technical report. Technical Report FHWA/TX-13/0-6670-1, Texas A&M Transportation Institute, 2012.

[136] Xiaoduan Sun, Yuebin Li, Dan Magri, and Hadi H. Shirazi. Application of *Highway Safety Manual* draft chapter: Louisiana experience. Transportation Research Record, 1950:55–64, 2006.

[137] Srinivasa Sunkari, Apoorba Bibeka, Nadeem Chaudhary, and Kevin Balke. Impact of traffic signal controller settings on the use of advanced detection devices. Technical Report FHWA/TX-18/0-6934-R1, Texas A&M Transportation Institute, 2019.

[138] Richard S. Sutton and Andrew G. Barto. Early history of reinforcement learning. In Reinforcement Learning: An Introduction, pages 11–17. The MIT Press, 2018.

[139] Meirav Taieb-Maimon and David Shinar. Minimum and comfortable driving headways: Reality versus perception. Human Factors, 43(1):159–172, 2001.

[140] Kai Liang Tan, Anuj Sharma, and Soumik Sarkar. Robust deep reinforcement learning for traffic signal control. Journal of Big Data Analytics in Transportation, 2(3):263–274, 2020.

[141] Keshuang Tang, Manfred Boltze, Zong Tian, and Hideki Nakamura. Initial comparative analysis of international practice in road traffic signal control. In Global Practices on Road Traffic Signal Control, pages 285–310. Elsevier, 2019.

[142] Leo Tasca. A review of the literature on aggressive driving research. Technical report, Ontario Ministry of Transportation, Toronto, 2000.

[143] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '17, pages 23–30, 2017.

[144] Ernest Peter Todosiev. The action point model of the vehicle-driver system. PhD thesis, The Ohio State University, Columbus, 1963.

[145] Tomer Toledo, Haris N. Koutsopoulos, and Moshe Ben-Akiva. Integrated driving behavior modeling. Transportation Research Part C: Emerging Technologies, 15(2):96–112, 2007.

[146] Elise van der Pol and Frans A. Oliehoek. Coordinated deep reinforcement learners for traffic light control. In Proceedings of the 30th Conference on Neural Information Processing Systems, NIPS '16, pages 1–8, 2016.

[147] Fadhely Viloria, Kenneth Courage, and Donald Avery. Comparison of queue-length models at signalized intersections. Transportation Research Record, 1710(1):222–230, 2000.

[148] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki amd Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 575:350–354, 2019.

[149] Katja Vogel. What characterizes a "free vehicle" in an urban area? Transportation Research Part F: Traffic Psychology and Behaviour, 5:15–29, 2002.

[150] Peter Wagner. Action point models of human driving behaviour. In Proceedings of the 2008 Traffic Simulation Workshop, Graz, 2008. Institute of Highway Engineering and Transport Planning, Graz University of Technology.

[151] Hao Wang, Yun Yuan, Xianfeng Terry Yang, Tian Zhao, and Yang Liu. Deep Q learning-based traffic signal control algorithms: Model development and evaluation with field data. Journal of Intelligent Transportation Systems, 2022.

[152] Min Wang, Libing Wu, Jianxin Li, and Liu He. Traffic signal control with reinforcement learning based on region-aware cooperative strategy. IEEE Transactions on Intelligent Transportation Systems, 2021.

[153] Yongheng Wang, Shaofeng Geng, and Qian Li. Intelligent transportation control based on proactive complex event processing. In Proceedings of the 3rd International Conference on Mechanics and Mechatronics Research, ICMMR '16, pages 1–5, 2016.

[154] Francis Rhys Ward and Ibrahim Habli. An assurance case pattern for the interpretability of machine learning in safety-critical systems. In Proceedings of the 2020 International Conference on Computer Safety, Reliability, and Security, SAFECOMP '20, pages 395–407, 2020.

[155] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. Machine Learning, 8:279–292, 1992.

[156] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 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, KDD '19, pages 1290–1298, 2019.

[157] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. CoLight: Learning network-level cooperation for traffic signal control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, pages 1913–1922, 2019.

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

[159] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. ACM SIGKDD Explorations Newsletter, 22(2):12–18, 2021.

[160] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. IntelliLight: A reinforcement learning approach for intelligent traffic light control. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, pages 2496–2505, 2018.

[161] Rainer Wiedemann. Simulation des straßenverkehrsflußes. Publications of the Institute for Transportation, University of Karlsruhe, 8:1–42, 1974.

[162] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman. Simulation and optimization of traffic in a city. In Proceedings of the 2004 Intelligent Vehicles Symposium, IV '04, pages 453–458, 2004.

[163] Lingtao Wu, Dominique Lord, and Yajie Zou. Validation of crash modification factors derived from cross-sectional studies with regression models. Transportation Research Record, 2514:88–96, 2015.

[164] Yuankai Wu, Huachun Tan, Lingqiao Qin, and Bin Ran. Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm. Transportation Research Part C: Emerging Technologies, 117:102649, 2020.

[165] Donghan Xie, Zhi Wang, Chunlin Chen, and Daoyi Dong. IEDQN: Information exchange DQN with a centralized coordinator for traffic signal control. In Proceedings of the 2020 International Joint Conference on Neural Networks, IJCNN '20, pages 1–8, 2020.

[166] Fei Xie, Kristie Gladhill, Karen K. Dixon, and Christopher M. Monsere. Calibration of *Highway Safety Manual* predictive models for Oregon state highways. Transportation Research Record, 2241:19–28, 2011.

[167] Xiao-Feng Xie, Gregory J. Barlow, Stephen F. Smith, and Zachary B. Rubinstein. Accounting for real-world uncertainty in real-time adaptive traffic control. Technical Report ATCSTR12, Carnegie Mellon University, 2012.

[168] Bingyu Xu, Yaowei Wang, Zhaozhi Wang, Huizhu Jia, and Zongqing Lu. Hierarchically and cooperatively learning traffic signal control. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI '21, pages 1–9, 2021.

[169] Ming Xu, Jianping Wu, Ling Huang, Rui Zhou, Tian Wang, and Dongmei Hu. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. Journal of Intelligent Transportation Systems, 24(1):1–10, 2020.

[170] Shantian Yang, Bo Yang, Zhongfeng Kang, and Lihui Deng. IHG-MA: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. Neural Networks, 139:265–277, 2021.

[171] Shantian Yang, Bo Yang, Hau-San Wong, and Zhongfeng Kang. Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm. Knowledge-Based Systems, 183:104855, 2019.

[172] Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A survey on reinforcement learning models and algorithms for traffic signal control. ACM Computing Surveys, 50(3):34, 2017.

[173] Biao Yin and Monica Menendez. A reinforcement learning method for traffic signal control at an isolated intersection with pedestrian flows. In Proceedings of the 19th COTA International Conference of Transportation Professionals, CICTP '19, pages 3123–3135, 2019.

[174] Bingquan Yu, Jinqiu Guo, Qinpei Zhao, Jiangfeng Li, and Weixiong Rao. Smarter and safer traffic signal controlling via deep reinforcement learning. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM '20, pages 3345–3348, 2020.

[175] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. ACM Computing Surveys, 55(1):1–36, 2023.

[176] Zheng Zeng. GraphLight: Graph-based reinforcement learning for traffic signal control. In Proceedings of the 6th International Conference on Computer and Communication Systems, ICCCS '21, pages 645–650, 2021.

[177] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In Proceedings of the 2019 World Wide Web Conference, WWW '19, pages 3620–3624, 2019.

[178] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In Handbook of Reinforcement Learning and Control, pages 321–384. Springer, 2021.

[179] Lun Zhang, Shan Jiang, and Zheng Wang. Schedule-driven signal priority control for modern trams using reinforcement learning. In Proceedings of the 17th COTA International Conference of Transportation Professionals, CICTP '17, pages 2122–2132, 2017.

[180] Yunchang Zhang and Jon Fricker. Investigating smart traffic signal controllers at signalized crosswalks: A reinforcement learning approach. In Proceedings of the 7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS '21, pages 1–6, 2021.

[181] Wenshuai Zhao, Jorge Pe na Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence, SSCI '20, pages 737–744, 2020.

[182] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. Learning phase competition for traffic signal control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, pages 1963–1972, 2021.

[183] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. Diagnosing reinforcement learning for traffic signal control. arXiv preprint, 2019.

[184] Guanjie Zheng, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li. Diagnosing reinforcement learning for traffic signal control. arXiv preprint, 1905.04716:1–10, 2019.

[185] Zuduo Zheng. Recent developments and research needs in modeling lane changing. Transportation Research Part B: Methodological, 60:16–32, 2014.

[186] Pengyuan Zhou, Tristan Braud, Ahmad Alhilal, Pan Hui, and Jussi Kangasharju. ERL: Edge based reinforcement learning for optimized urban traffic light control. In Proceedings of the 3rd International Workshop on Smart Edge Computing and Networking, SmartEdge '19, pages 849–854, 2019.