Flow: A Modular Learning Framework for Autonomy in Traffic

Cathy Wu*, Abdul Rahman Kreidieh†, Kanaad Parvate‡, Eugene Vinitsky§, Alexandre M Bayen†‡¶
*MIT, Laboratory of Information and Decision Systems; MIT, Department of Civil and Environmental Engineering; MIT, Institute of Data Systems and Society
†UC Berkeley, Department of Civil and Environmental Engineering
‡UC Berkeley, Department of Electrical Engineering and Computer Sciences
§UC Berkeley, Department of Mechanical Engineering
¶UC Berkeley, Institute for Transportation Studies

***Abstract*—The rapid development of autonomous vehicles (AVs) holds vast potential for transportation systems through improved safety, efficiency, and access to mobility. However, new methodologies are needed for the design of vehicles and transportation systems to enable these positive outcomes, due to numerous technical, political, and human factors challenges. This article focuses on tackling important technical challenges arising from the *partial* adoption of autonomy (hence termed *mixed autonomy*, to involve both AVs and human-driven vehicles): partial control and partial observation, complex dynamics of multi-vehicle interactions, and the sheer variety of traffic settings represented by real-world networks. To enable the study of the full diversity of traffic settings, we first propose decomposing traffic control tasks into components called modules, which may be configured and composed to create new control tasks of interest. These modules include salient aspects of traffic control tasks: networks, actors, control laws, metrics, initialization, and additional dynamics. Second, we study the potential of model-free deep Reinforcement Learning (RL) methods to address the complexity of traffic dynamics. The resulting modular learning framework is called Flow. Using Flow, we create and study a variety of mixed-autonomy settings, including single-lane, multi-lane, and intersection traffic In all cases, the learned control law exceeds human driving performance (measured by system-level average velocity) by at least 40% with only 5-10% adoption of AVs. In the case of partially-observed single-lane traffic, we show that a low-parameter neural network control law can eliminate commonly observed stop-and-go waves in traffic. In particular, the control laws surpass all known model-based controllers to achieve near-optimal performance across a wide spectrum of vehicle densities (even with a memoryless control law) and additionally generalize to out-of-distribution vehicle densities.**

***Index Terms*—Deep Reinforcement Learning; Traffic Microsimulation; Autonomous Vehicles**

I. Introduction

Autonomous vehicles (AVs) are projected to enter our society in the very near future, with full adoption in select areas expected as early as 2050 [1]. A recent study further estimated that fuel consumption in the U.S. could decrease as much as 40% or *increase* as much as 100% once autonomous fleets of vehicles are rolled out onto the streets [1], potentially exacerbating the 28% of energy consumption that is already due to transportation in the US [2]. These factors include incorporation of platooning and eco-driving practices, vehicle right-sizing, induced demand, travel cost reduction, and new mobility user groups. As such, computational tools are needed with which to design, study, and control these complex large-scale robotic systems.

Corresponding author: Cathy Wu (cathywu@mit.edu)
Email addresses: {aboudy, kanaad, evinitsky, bayen}@berkeley.edu

Existing tools are largely limited to commonly studied cases where AVs are few enough to not affect the environment [3], [4], [5], such as the surrounding traffic dynamics, or so many as to reduce the problem to primarily one of coordination [6], [7], [8]. For clarity, we refer to these as the *isolated autonomy* and *full autonomy* cases, respectively. At the same time, the intermediate regime, which is the long and arduous transition from no (or few) AVs to full adoption, is poorly understood. We term this intermediate regime *mixed autonomy*. The understanding of mixed autonomy is crucial for the design of suitable vehicle controllers, efficient transportation systems, sustainable urban planning, and public policy in the advent of AVs. This article focuses on autonomous vehicles, which we expect to be among the first of robotic systems to enter and widely affect existing societal systems. Additional highly anticipated robotic systems, which may benefit from similar techniques as presented in this article, include aerial vehicles, household robotics, automated personal assistants, and additional infrastructure systems.

Motivated by the importance and uncertainty about the *transition* in the adoption of AVs, this article deviates from focus in the literature on isolated and full autonomy settings and proposes the mixed autonomy setting. The mixed autonomy setting exposes heightened complexities due to the interactions of many human and robotics agents in highly varied contexts, for which the predominant model-based approaches of the traffic community are largely unsuitable. Instead, we posit that model-free deep reinforcement learning (RL) allows us to de-couple mathematical modeling of the system dynamics and control law design, thereby overcoming the limitations of classical approaches to autonomous vehicle control in complex environments. Specifically, we propose a modular learning framework, in which environments representing complex control tasks are comprised of reusable components. We validate the proposed methodology on widely studied traffic benchmark exhibiting backward propagating traffic shockwaves in a partially-observed environment, and

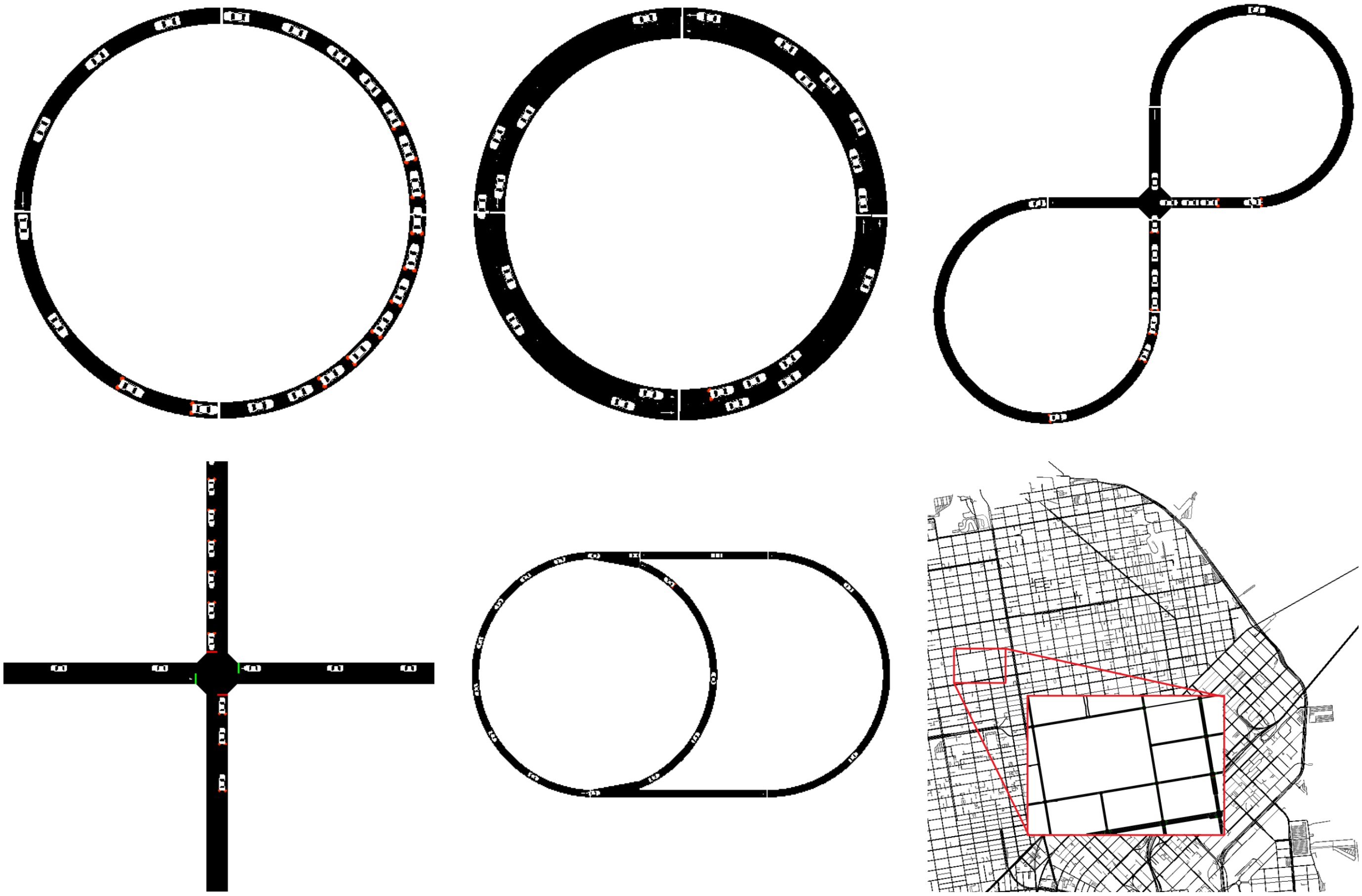

Fig. 1: Example network modules supported by the Flow framework. **Top left**: Single-lane circular track. **Top middle**: Multi-lane circular track. **Top right**: Figure-eight road network. **Bottom left**: Intersection network. **Bottom middle**: Closed loop merge network. **Bottom right**: Imported San Francisco network. In Flow, scenarios can be generated using OpenStreetMap (OSM)[1]data and vehicle dynamics models from SUMO.

we subsequently produce a control law which far exceeds all previous methods, generalizes to unseen traffic densities, and closely matches control theoretic performance bounds. By appropriately composing the reusable components, we further demonstrate the effectiveness of the methodology with preliminary results on more complex traffic scenarios for which control theoretic results are not known. In order to facilitate future research in mixed autonomy traffic, we developed and open-sourced the modular learning framework as Flow[2], which exposes design primitives for composing traffic control tasks. Our contributions aim to enable the community to study not only mixed autonomy scenarios which are composites of analytically tractable mathematical frameworks, but also arbitrary networks, or even full-blown traffic microsimulators, designed to simulate hundreds of thousands of agents in complex environments (see examples in Figure 1).

The contributions of Flow to the research community are multi-faceted. For the robotics community, Flow seeks to enable rich characterizations and empirical study of complex, large-scale, and realistic multi-robot control tasks. For the machine learning community, Flow seeks to expose to modern reinforcement learning algorithms a class of challenging control tasks derived from an important real-world domain. For the control community, Flow seeks to provide intuition, through successful learned control laws, for new provable control techniques for traffic-related control tasks. Finally, for the transportation community, Flow seeks to provide a new pathway, through reusable traffic modules and modern reinforcement learning methods, to addressing new challenges concerning AVs and long-standing challenges concerning traffic control.

The rest of the article is organized as follows: Section II introduces the problem of mixed autonomy. Section III presents related work to place this article in the broader context of automated vehicles, traffic flow modeling, and deep RL. Section IV summarizes requisite concepts from RL and traffic. Section V describes the modular learning framework for the scalable design and experimentation of traffic control tasks. This is followed by two experimental sections: Section VI, in which we validate the modular learning framework on a canonical traffic control task, and Section VII which presents more sophisticated applications of the framework to more complex traffic control tasks.

[2]Our modular learning framework called Flow is open-source and available at `https://github.com/flow-project/flow`.

II. Mixed autonomy

To make headway towards understanding the complex integration of AVs into existing transportation and urban systems, this article introduces the problem of mixed autonomy to frame the study of partial adoption of autonomy into an existing system. Although mixed autonomy technically subsumes isolated and full autonomy, both are still important and active areas of research and will often lead to more efficient algorithms for specialized settings. However, we observe that there are important robotics problems that are not addressed by either body of approaches, and thus formulate the problem of mixed autonomy traffic and propose a suitable methodology.

For a particular type of autonomy (e.g. AVs, traffic lights, roadway pricing), we define *mixed autonomy* as the intermediate regime between a system with no adoption of autonomy and a system where the autonomy is employed fully. For example, in the context of AVs, full autonomy corresponds to 100% vehicles being autonomously driven, so mixed autonomy corresponds to any fraction of vehicles being autonomously driven. In the context of traffic lights, full autonomy corresponds to all traffic lights having automated signal control, so mixed autonomy corresponds to any fraction of traffic lights being automated; the other lights could follow some different pre-programmed rules or could not be utilized at all. As the examples allude, mixed autonomy must be defined alongside its boundary conditions of no autonomy and full autonomy as well as its system context. For example, the full system context could be a segment of roadway, a full road, or a entire urban region. Mixed autonomy differs from the previously studied cases in two important ways.

First, the evaluation criteria for mixed autonomy settings will typically be less clear than in studies of isolated autonomy. Studies in isolated autonomy are often evaluated with respect to known human or expert performance for a similar task. For example, the performance of a self-driving vehicle in the isolated autonomy context is typically measured against that of a human driver [9], [10]. Similarly, expert demonstrations are often considered the objective in robot learning for a variety of tasks, including locomotion, grasping, and manipulation [11], [12], [13]. In these cases, we may assume knowledge of a good control law and that it is feasible to attain, e.g. from a human or expert demonstrator. However, evaluating with respect to human performance makes implicit assumptions about the capabilities of the autonomous system and the optimality of human performance, both of which may be incorrect. For example, in the context of a mixed autonomy traffic system, evaluating with respect to the known human performance is restrictive; it is well known that human driving behavior induces (suboptimal) stop-and-go traffic in a wide regime of traffic scenarios [14], [15]. In mixed autonomy settings, we are instead interested evaluating autonomy with respect to a different, possibly system-level objective, in order to understand its potential effects on the system.

Second, the strict coupling between the mathematical modeling and the control law evaluation is a critical restriction for studying mixed autonomy. Differently from isolated autonomy, full autonomy is indeed often evaluated with respect to a system-level objective. However, its system dynamics are often much simpler than that of mixed autonomy. In full autonomy settings, all autonomous components are known and need not be modeled, and uncertainty from human behavior is largely eliminated. Even so, full autonomy is far from solved; however, numerous full autonomy settings can be analyzed using techniques from control theory, in particular when the system dynamics may be modeled within a number of powerful mathematical frameworks, including partial differential equations [16], [17], [18], [19], ordinary differential equations [20], [21], [22], [23], queuing systems [24], [25], [26], etc. In a number of cases, control theoretic performance bounds can be analytically derived or an optimal controller may even be found directly. In contrast, mixed autonomy suffers from additional challenges including interactions with humans of unknown or complex dynamics, partial observability from sensing limitations, and partial controllability due to the lack of full autonomy. However, these control theoretic performance bounds be employed as reference points when evaluating mixed autonomy systems. For mixed autonomy settings, these aspects are often prohibitively complex to characterize within a single mathematical framework.

In this work, we propose and validate a new methodology for addressing mixed autonomy traffic. We posit that sampling-based optimization allows us to de-couple mathematical modeling of the system dynamics and control law design for arbitrary evaluation objectives, thereby overcoming the limitations of studies in both isolated and full autonomy. In particular, we propose that model-free deep reinforcement learning (RL) is a compelling and suitable framework for the study of mixed autonomy. The decoupling allows the designer to specify arbitrary control objectives and system dynamics to explore the effects of autonomy on complex systems. For the system dynamics, the designer may model a system of interest in whichever mathematical or computational framework she wishes, and we require only that the model is consistent with a (Partially Observed) Markov Decision Process, (PO)MDP, interface. For control law design, deep neural network architectures may be used for representing large and expressive control law (or policy) classes. Finally, the resulting framework employs model-free deep RL to enable the designer to explore the effects of autonomy on a complex system, up to local optimality with respect to the control law parameterization.

III. Related Work

Control of automated vehicles. Automated and autonomous vehicles have been studied in a myriad of contexts, which we frame in the context of isolated, full, and mixed autonomy.

Isolated autonomy. Spurred by the US DARPA challenges in autonomous driving in 2005 and 2007 [27], [4], countless efforts have demonstrated the increasing ability of vehicles to operate autonomously on real roads and traffic conditions, without custom traffic infrastructure. These vehicles instead rely largely on sensors (LIDAR, radar, camera, GPS), computer vision, motion planning, mapping, and behavior prediction, and are designed to obey traffic rules. Robotics has continued to demonstrate tremendous potential in improving transportation systems through AVs research; highly related problems include localization [28], [29], [30], path planning [31], [32], collision avoidance [33], and perception [34]. Considerable progress has also been made in recent decades in vehicle automation, including anti-lock braking systems (ABS), adaptive cruise control (ACC), lane keeping, lane changing, parking, overtaking, etc. [35], [36], [37], [38], [39], [40], [41], [42], [43], which also have great potential to improve safety and efficiency in traffic. The development of these technologies are currently focused on the performance of the individual vehicle, rather than its interactions or effects on other parts of the transportation system.

Full autonomy. At the other end of the autonomy spectrum, all vehicles are automated and operate efficiently with collaborative control. Model-based approaches to such *full autonomy* have permitted the reservation system design and derivation of vehicle weaving control for fully automated intersections [6], [44] and roads [7].

Mixed autonomy. A widely deployed form of vehicle automation is adaptive cruise control (ACC), which adjusts the longitudinal dynamics for vehicle pacing and driver comfort, and is grounded in classical control techniques [45], [46], [47]. Similarly, cooperative ACC (CACC) uses control theory to simultaneously optimize the performance of several adjacent vehicles, such as for minimizing the fuel consumption of a vehicle platoon [48], [49], [50], [51], [36]. ACC/CACC may be viewed as an early form of mixed autonomy. This article will similarly study longitudinal vehicle controllers, with a key difference being our focus on system-level rather than local objectives, such as the average velocity of all vehicles in the system, which is typically important in operations and planning contexts.

Recently, a few studies have started to use formal techniques for controller design for system-level evaluation of mixed autonomy traffic, including state-space [52] and frequency domain analysis [108]. There are also several modeling- and simulation-based evaluations of mixed autonomy systems [54], [55], [56] and model-based approaches to mixed autonomy intersection management [57]. Despite these advances in controller design, these approaches are generally limited to simplified models, such as homogeneous, non-delayed, deterministic driver models, or restricted network configurations. This article proposes to overcome these barriers through model-agnostic sampling-based methods.

Recent field operational tests by Stern, et al.[58] demonstrated a reduction in fuel consumption of 40% by the insertion of an autonomous vehicle in traffic in a circular track to dampen the famous instabilities displayed by Sugiyama, et al. [14] in their seminal experiment. These tests motivate the present work: it demonstrates the power of automation and its potential impact on complex traffic phenomena such as *stop-and-go* waves [59].

On a large-scale network, fleets of AVs have recently been explored in the context of shared-mobility systems, such as autonomous mobility-on-demand systems [60], [5], [61], which abstracts out the low-level vehicle dynamics and considers a queuing theoretic model. Low-level vehicle dynamics, however, are crucial, as exhibited by Sadigh, et al. [62] and because many traffic phenomena exhibited at the level of low-level dynamics affect energy consumption, safety, and travel time [14], [63], [64], [65].

Modeling and control of traffic. Mathematical modeling and analysis of traffic dynamics is notoriously complex and yet is a prerequisite for traffic control [101], [67], regardless of if the control input is a vehicle, traffic light, ramp meter, or toll. Such mathematical frameworks include partial differential equations, ordinary differential equations (ODEs), queuing systems, stochastic jump systems, for the modeling of highway traffic, longitudinal dynamics, intersections, and lateral dynamics, respectively. Researchers have traded the complexity of the model (and thus its realism) for the tractability of analysis, with the ultimate goal of designing optimal and practical controllers. Consequently, results in traffic control can largely be classified as simulation-based numerical analysis with rich models but minimal theoretical guarantees [68], [69], [70], [71], or theoretical analysis on simplified models such as assuming non-oscillatory responses [72] or focusing on a single-lane circular track [15], [73], [74], [7], [75], [108].

Analysis techniques are often tightly coupled with the mathematical framework. For instance, linear systems theory techniques may be used with ODEs and stochastic processes may be used with queuing systems, but they may be incompatible with other mathematical frameworks. Thus, there are virtually no theoretical works that simultaneously study lateral dynamics, intersection dynamics, longitudinal dynamics, etc., for the reason that they are typically all modeled using different mathematical frameworks. As a notable exception, the work of Miculescu and Karaman [44] takes a model-based approach which considers both intersections and longitudinal dynamics for simplified fully-automated intersection. This article seeks to take a step towards decoupling the reliance of control from the mathematical modeling of the problem, by proposing abstractions which permit the composition of reusable modules to specify problems and optimize for locally optimal controllers.

Deep reinforcement learning. Deep RL, which is the key workhorse in our framework, is a powerful tool for control and has demonstrated success in complex, data-rich problems such as Atari games [76], 3D locomotion and manipulation [77], [78], [79], among others. These recent advances in deep RL provide a promising alternative to model-based controller design. In a broader context, the use of RL for mixed autonomy traffic may be viewed as an intermediate step towards eventual deployment, informing the designer of the vehicle controller or the designer of the transportation system about a variety of complex performance metrics.

RL testbeds exist for different problem domains, such as the *Arcade Learning Environment* (ALE) for Atari games [80], DeepMind Lab for a first-person 3D game [81], OpenAI gym for a variety of control problems [82], FAIR TorchCraft for Starcraft: Brood War [83], MuJoCo for multi-joint dynamics with Contact [84], TORCS for a car racing game [85], among others. Each of these RL testbeds enables the study of control through RL of a specific problem domain by leveraging the data-rich setting of simulation. Although task design and specification is a key part of reinforcement learning research, relatively little attention has been given to the systematic construction or configuration of RL tasks. A notable exception is TextWorld [86], which programmatically generates a range of text-based adventure games. One of the primary goals of this article is to present a framework for enabling configurable RL tasks in studying mixed autonomy traffic.

Deep RL and Traffic. Several recent studies incorporated ideas from deep learning in traffic optimization. Deep RL has been used for traffic prediction [87], [88] and control [89], [90], [91]. A deep RL architecture was used by Polson, et al. [87] to predict traffic flows, demonstrating success even during special events with nonlinear features; to learn features to represent states involving both space and time, Lv, et al. [88] additionally used hierarchical autoencoding for traffic flow prediction. Deep Q Networks (DQN) was employed for learning traffic signal timings in Li, et al. [89]. A multi-agent deep RL algorithm was introduced in Belletti, et al. [90] to learn a control law for ramp metering. Wei, et al. [92], [91] employs reinforcement learning and graph attention networks for control of traffic signals. For additional uses of deep learning in traffic, we refer the reader to Karlaftis, et al. [93], which presents an overview comparing non-neural statistical methods and neural networks in transportation research. These recent results demonstrate that deep learning and deep RL are promising approaches to traffic problems. This article is the first to employ deep RL to design controllers for AVs and assess their impacts in traffic. A preliminary prototype of our architecture is published [94] and an earlier version of this manuscript is available [95].

IV. PRELIMINARIES

In this section, we define the notation and key concepts used in subsequent sections.

A. Markov Decision Processes

The system described in this article solves tasks which conform to the standard interface of a finite-horizon discounted *Markov decision process* (MDP) [96], [97], defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma, T)$, where $\mathcal{S}$ is a (possibly infinite) set of states, $\mathcal{A}$ is a set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is the transition probability distribution, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\rho_0 : \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is the initial state distribution, $\gamma \in (0, 1]$ is the discount factor, and T is the time horizon. For partially observable tasks, which conform to the interface of a *partially observable Markov decision process* (POMDP), two more components are required, namely Ω, a set of observations, and $\mathcal{O} : \mathcal{S} \times \Omega \rightarrow \mathbb{R}_{\geq 0}$, the observation probability distribution.

B. Reinforcement learning

RL studies the problem of how agents can learn to take actions in its environment to maximize its cumulative reward. The article uses policy gradient methods [98], a class of reinforcement learning algorithms which optimize a stochastic policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_{\geq 0}$. Although commonly called a policy, we will generally refer to π as a controller or control law in this article, to be consistent with the phrasing of traffic control. These algorithms iteratively update the parameters of the control law through optimizing the expected cumulative reward using sampled data from a traffic simulator or model. The control law is commonly a neural network. Two control laws used in this article are the *Multilayer Perceptron* (MLP) and *Gated Recurrent Unit* (GRU). The MLP is a classical artificial neural network with multiple hidden layers and utilizes backpropagation to optimize its parameters [99]. The GRU is a recurrent neural network capable of storing memory on the previous states of the system [100]. GRUs make use of parameterized update and reset gates, which enable them to make decisions based on both current and past inputs and are also optimized by the policy gradient method.

C. Vehicle dynamics models

The environments studied in this article are traffic systems. Basic traffic dynamics on single-lane roads can be represented by *ordinary differential equation* (ODE) models known as *car following models* (CFMs). These models describe the longitudinal dynamics of human-driven vehicles, given only observations about itself and the vehicle preceding it. CFMs vary in terms of model complexity, interpretability, and their ability to reproduce prevalent traffic phenomena, including stop-and-go traffic waves. For modeling of more complex traffic dynamics, including lane changing, merging, driving near traffic lights, and city driving, we refer the reader to the text of Treiber and Kesting [101] dedicated to this topic.

Standard CFMs are of the form:

$$a_i = \dot{v}_i = f(h_i, \dot{h}_i, v_i), \tag{1}$$

where the acceleration a_i of car i is some typically nonlinear function of $h_i, \dot{h}_i, v_i$, which are the headway, relative velocity, and velocity for vehicle i, respectively. Though a general model may include time delays from the input signals $h_i, \dot{h}_i, v_i$ to the resulting output acceleration a_i, we will consider a non-delayed system, where all signals are measured at the same time instant t. Example CFMs include the Intelligent Driver Model (IDM) [21] and the Optimal Velocity Model (OVM) [22], [23].

D. Intelligent Driver Model

The *Intelligent Driver Model* (IDM) is a car following model capable of accurately representing realistic driver behavior [21] and reproducing traffic waves, and is commonly used in the transportation research community. We will employ IDM in the numerical experiments of this article, and therefore analyze this specific model to compute the theoretical performance bounds of the overall traffic system. The acceleration for a vehicle modeled by IDM is defined by its bumper-to-bumper headway h (distance to preceding vehicle), velocity v, and relative velocity $\dot{h}$, via the following equation:

$$a_{\text{IDM}} = \frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{H(v, \dot{h})}{h}\right)^2\right] \tag{2}$$

where $H(\cdot)$ is the desired headway of the vehicle, denoted by:

$$H(v, \Delta v) = s_0 + \max\left(0, vT + \frac{v\dot{h}}{2\sqrt{ab}}\right) \tag{3}$$

where $h_0, v_0, T, \delta, a, b$ are given parameters. Table I describes the parameters of the model and provides typically used values [101].

	Intelligent Driver Model (IDM)						
Parameter	v_0	T	a	b	δ	h_0	noise
Value	30 m/s	1 s	1 m/s^2	1.5 m/s^2	4	2 m	$\mathcal{N}(0, 2)$

TABLE I: Parameters for car-following control law

E. Traffic, equilibria, and performance bounds

The overall traffic system is comprised of interconnected dynamical models, including CFMs, representing the many interacting vehicles in the traffic system. An example of a traffic system which can be represented by only longitudinal vehicle dynamics is a circle of n vehicles, all driving clockwise, and each of which is following the vehicle preceding it. The result is a dynamical system which consists of a cascade of n potentially nonlinear and delayed vehicle dynamics models. This example will serve as our first numerical experiment (Section VI), as it is able to reproduce important traffic phenomena (traffic waves and traffic jams), and, despite its simplicity, its optimal control problem in the mixed autonomy setting has remained an open research question.

Although the primary methodology explored in this article is model-agnostic, we employ control theoretic model-based analysis to compute the performance bounds of the traffic system, so that we can adequately assess the performance of the learned control laws. In a homogeneous setting, where all vehicles follow the same dynamics model, *uniform flow* describes the situations where all vehicles moves at some constant velocity v^* and constant headway h^*, and corresponds to one of the equilibria of the traffic system. It can be represented in terms of the vehicle dynamics model by:

$$a_i = 0 = f(h^*, 0, v^*). \tag{4}$$

This equation defines the relationship between the two equilibrium quantities h^*, v^* for a general car following model, and the specific relationship for IDM is displayed in Figure 3 (green curve). Uniform flow equilibria correspond to high velocities, and are thus desirable, but these equilibria are also unstable due to properties of human driving behavior [102].

It is intuitive to think of the equilibrium density (which is inversely related to the equilibrium headway h^*) as a traffic condition. We will seek to evaluate our learned control laws against a range of traffic conditions. Each traffic condition (density) has associated with it an optimal equilibrium velocity v^*. In practice, the equilibrium density can be approximated by the local traffic density. In settings with heterogeneous vehicle types, the equilibrium can be numerically solved by constraining the total headways to be the total road length and the velocities to be uniform. It is important to note the difference between the equilibrium velocity v^* and the target velocity v_0 (free flow speed) of the vehicle models; v_0 can be thought of as a speed limit for highway traffic [21]. On the other hand, v^* is a control theoretic quantity jointly determined by the traffic condition h^*, the target velocity v_0, the system dynamics, and various other parameters.

Another set of equilibria exhibited by the example traffic system corresponds to traffic waves (also called stop-and-go waves). These traffic waves are stable limit cycles, that is, closed curves that trajectories tend towards (rather than away) [15]. In other words, the typical traffic system tends towards exhibiting traffic waves. We view this as a performance lower bound for the mixed autonomy setting because any control law which yields worse performance could be replaced by a human driver to yield a better outcome. Using IDM, the relationship between density and the average velocity with traffic waves is displayed in Figure 3 (red curve).

Finally, we are concerned with *steady-state performance* of the traffic system, rather than any instantaneous performance. We thus take the uniform flow and traffic waves equilibria to describe the upper and lower bounds, respectively, for steady-state performance of the overall traffic system. We note that, because we are analyzing the *no autonomy* setting, but evaluating against a *mixed autonomy* setting, the performance bounds presented here should be viewed as close approximations to the true performance bounds. For more detailed performance bounds, we refer the reader to related work [15], [103], [104].

V. FLOW: A MODULAR LEARNING FRAMEWORK

While the research community in deep RL focuses on solving pre-defined tasks, relatively little emphasis is placed on efficient scenario creation (see Section III). We use the term **scenario** to describe the full traffic setting, including all learning and non-learning components, and it conforms to a (PO)MDP interface (see Section IV). In robotics, the term "task" is commonly used; we prefer the word "scenario" to also capture settings with no learning components or no goal to achieve (e.g., situations with only fixed human driver models).

In large-scale multi-agent systems such as traffic, the ability to efficiently study a wide range of scenarios is especially important due to the highly varied and complex nature of settings, including the number of vehicles, heterogeneity of agents, types of network configurations, regional behaviors and regulations, etc. To this end, this article contributes an approach that decomposes a scenario into modules which can be configured and composed to create new scenarios of interest. Flow is the resulting modular learning framework for enabling the creation, study, and control of complex traffic scenarios.

A. Scenario modules

Flow is comprised of the following modules, which can be assembled to form scenarios of interest (see Figure 2).

Network: The network specifies the physical road layout, in terms of roads, lanes, length, shape, roadway connections, and additional attributes. Example include a two-lane circular track with circumference 200m, or the structure described by importing a map from OpenStreetMap (see Figure 1 for examples

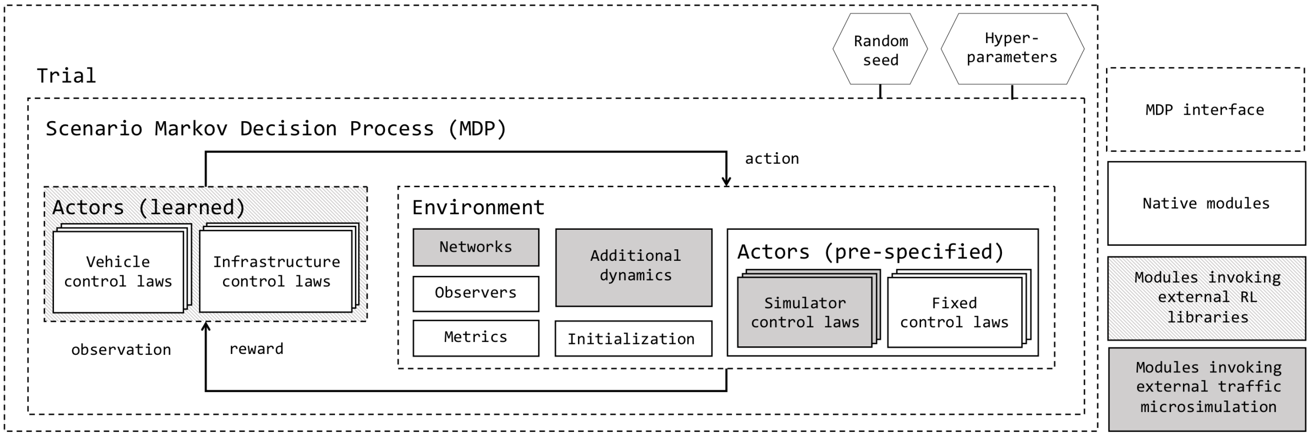

Fig. 2: Flow overview. Flow is a modular learning framework, which enables the study of diverse mixed autonomy traffic scenarios with deep reinforcement learning (RL) methods. Scenarios are defined within the Markov Decision Process (MDP) framework. The workhorse of Flow is modules (rectangles) which can be composed to comprise scenarios, including networks, actors, observers, control laws, metrics, initializations, and additional dynamics. Actors may refer to vehicles or infrastructure, and may be learned or pre-specified. Additional dynamics may include vehicle fail-safes, right-of-way rules, and physical limitations. Additional parameters (hexagons) may also be configured. Flow facilitates the composition of these modules to create new traffic scenarios of interest, which may then be studied as traffic control problems using deep RL methods. Flow invokes external libraries for training and simulation.

of supported networks). Several of these examples will be used in demonstrative experiments in Sections VI and VII. More details about the specific networks are in Appendix A.

Actors: The actors describe the physical entities in the environment which issue control signals to the environment. In contrast to isolated autonomy settings, in a traffic setting, there are typically many interacting physical entities. Due to the focus on mixed autonomy, this article specifically studies vehicles as its physical entities. Other possible actors may include pedestrians, bicyclists, traffic lights, roadway signs, toll booths, as well as other transit modes and infrastructure.

Observer: The observer describes the mapping $\mathcal{S} \to \mathcal{O}$ and yields the function of the state that is observed by the actor(s). The output of the observer is taken as input to the control law, described below. For example, while the state may include the position, lane, velocity, and accelerations of all vehicles in the system, the observer may restrict access to only information about local vehicles and aggregate statistics, such as average speed or length of queue at an intersection.

Control laws: Control laws dictate the behaviors of the actors and are functions mapping observations to control inputs $\mathcal{O} \to \mathcal{A}$. All actors require a control law, which may be pre-specified or learned. For instance, a control law may represent a human driver, an autonomous vehicle, or even a set of vehicles. That is, a single control law may be used to control multiple vehicles in a *centralized control* setting. Alternatively, a single control law may be used by multiple actors in a *shared parameter control* setting.

Dynamics: The dynamics module consists of additional submodules which describe different aspects of the system evolution, including vehicle routes, demands, stochasticity, traffic rules (e.g., right-of-way), and safety constraints.

Metrics: The metrics describe pertinent aggregated statistics of the environment. The reward signal for the learning agent is a function of these metrics. Examples include the overall (average) velocity of all vehicles and the number of (hard) braking events.

Initialization: The initialization describes the initial configuration of the environment at the start of an episode. Examples include setting the position and velocity of vehicles according to different probability distributions.

Sections VI and VII demonstrate the potential of the framework. Whereas in a model-based framing, many modules are simply not re-configurable due to differences in the mathematical descriptions (e.g. discrete versus continuous control inputs, such as in the case of longitudinal and lateral control), in this model-agnostic framework, disparate dynamics may be captured in the same scenario and effectively studied using sampling-based optimization techniques such as deep reinforcement learning.

B. Architecture and implementation

The implementation of Flow is open source and builds upon open source software to promote access and extension. The open source approach further aims to support the further development of custom modules, to permit the study of richer and more complex environments, agents, metrics, and algorithms. The implementation builds upon SUMO (**S**imulation of **U**rban **MO**bility) [105] for vehicle and traffic modeling, Ray RLlib [106], [107] for reinforcement learning, and OpenAI gym [82] for the MDP.

SUMO is a microscopic traffic simulator, which explicitly models vehicles, pedestrians, traffic lights, and public transportation. It is capable of simulating urban-scale road networks. SUMO has a rich Python API called TraCI (**Tra**ffic **C**ontrol **I**nterface). Ray RLlib are frameworks that enable training and evaluating of RL algorithms on a variety of scenarios, from classic tasks such as cartpole balancing to more complicated tasks such as 3D humanoid locomotion. OpenAI gym is an MDP interface for reinforcement learning tasks.

Flow is implemented as a lightweight architecture to connect and permit experimentation with the modules described in the previous section. As typical in reinforcement learning studies, an **environment** encodes the MDP. The environment facilitates the composition of dynamics and other modules, stepping through the simulation, retrieving the observations, sampling and applying actions, computing the metrics and reward, and resetting the simulation at the end of an episode. A **generator** produces network configuration files compatible with SUMO according to the network description. The generator is invoked by the experiment upon initialization and, optionally, upon reset of the environment, allowing for a variety of initialization conditions, such as sampling from a distribution of vehicle densities. Flow then assigns control inputs from the different control laws to the corresponding actors, according to an **action assigner**, and uses the TraCI library to apply actions for each actor. Actions specified as accelerations are converted into velocities, using numerical integration and based on the timestep and current state of the experiment.

Finally, Flow is designed to be inter-operable with classical model-based control methods for evaluation purposes. That is, the learning component of Flow is optional, and this permits the fair comparison of diverse methods for traffic control.

VI. Configurable modules for mixed autonomy

This section demonstrates that Flow can achieve high performance in a challenging and classic traffic congestion scenario.

A. Background

Although single-lane traffic has been studied over several decades in many variants, including closed networks [14], [15], [52], [104], open networks [46], [68], [108], different human driving models [101], and different objectives [54], previous works have focused on modeling the problem, characterizing the system dynamics, and optimizing for local performance (e.g. comfort). Relatively little work has been able to broach the problem of how to address undesirable traffic congestion through the control of vehicles. To the best of the authors knowledge, no prior work has achieved near optimality in the mixed autonomy setting for single-lane traffic congestion (in the form considered in this article or its many variants). Two related works of note, which study the same traffic scenarios, include Stern, et al. [58] and Horn, et al. [7]. Respectively, these works do not achieve near-optimality and do not consider the mixed autonomy setting. Importantly, both works carefully devise a control law based on knowledge of the environment, and thus we refer to them as model-based control laws. Additionally, eco-driving practices provide heuristic guidance to drivers to ease traffic congestion [109], [110]. In contrast, the approach proposed by this work requires a significantly lighter form of "design supervision," in the form of a reward function, which largely avoids explicitly employing domain knowledge or mathematical analysis.

Specifically, this section studies the proposed methodology applied to the mixed autonomy single-lane circular track, following the canonical setup [14], which consists of 22 human-driven vehicles on a 230 m circular track. This seminal experiment shows that such a dynamical system produces backward propagating waves, causing part of the traffic to come to a complete stop, even in the absence of typical traffic perturbations, such as those caused by lane changes, merges or stop lights. The field experiment of Stern, et al. [58] studies the mixed autonomy setting of 21 human-driven vehicles and one vehicle employing one of two proposed AV control laws, which we detail in Appendix B and compare with as baseline methods. This setting invokes a cascade of nonlinear dynamics from n (homogeneous) agents. In this and following sections, we study the potential of RL to produce well-performing control laws, in highly nonlinear and complex settings.

[2] OpenStreetMap: `https://www.openstreetmap.org/`

Experiment parameters	Value
simulation step	0.1 s/step
circular track range (train)	[220, 270] m
circular track range (test)	[210, 290] m
warmup time	75 s
time horizon	300 s
total number of vehicles	22
number of AVs	1

TABLE II: Network and simulation parameters for mixed autonomy circular track (single-lane).

B. Experiment Modules

We design the following experiment composed of the modules proposed in Section V-A. The network and simulation-specific parameters of our numerical experiments are summarized in Table II.

Network: We train on a set of networks consist of single-lane circular tracks, with lengths ranging between 220 m and 270 m (uniformly sampled), as displayed in Figure 1 (top left), in order to represent a continuous and wide range of traffic densities. An alternative approach (not considered here) to represent a range of densities is to fix the track length and instead vary the number of vehicles.

Actors: There are 22 vehicles, each 5 m long.

Observer: The observer maps the full state to only the velocity of the autonomous vehicle, the relative velocity to its preceding vehicle, and its relative position to the preceding vehicle.

Control laws: Separate control laws are used for the behavior of human drivers and AVs. Of the 22 actors, 21 are modeled as human drivers using the widely used *Intelligent Driver Model* (IDM) [101], which is included in Section IV and parameters detailed in Table I. When all 22 are modeled using human driver models, traffic jams are exhibited (see left side of Figures 4 and 5), similarly to Sugiyama, et al. [14], and thereby serves as a *no autonomy* baseline for comparison.

Due to the sampling-based nature of the proposed methodology, the performance of previous methods may be readily assessed relative to the RL approach. This provides a type of backwards compatibility with the research community, and permits the fair comparison of control laws, regardless of the underlying mathematical framework. We compare the following control laws for the single autonomous vehicle. Recall that all actors are partially observed.

- Learned control laws. We consider control laws with and without memory.
  - GRU control law (with memory): hidden layers (5,) and `tanh` non-linearity.
  - MLP control law (no memory): diagonal Gaussian MLP, two-layer network with hidden layer of shape (3,3) and `tanh` non-linearity.
- Model-based control laws
  - Proportional Integral (PI) control law with saturation, given in Stern, et al. [58] and is included in Section B-2 for completeness.
  - *FollowerStopper* control law, with desired velocity fixed at 4.15 m/s. The *FollowerStopper* control law is introduced in Stern, et al. [58] and is also detailed in Section B-1. *FollowerStopper* requires an external desired velocity, so we selected the largest fixed velocity which successfully mitigates stop-and-go waves at a track length of 260 m; this is further discussed in the results.

Dynamics: The IDM dynamics are additionally perturbed by Gaussian acceleration noise of $\mathcal{N}(0, 0.2)$, calibrated to match measures of stochasticity to the IDM model presented by Treiber, et al. [111]. For the first 75 seconds of the 300 second episode, the acceleration of the autonomous vehicle is overridden by the IDM model to ensure the formation of stop-and-go waves.

Metrics: We consider two natural metrics: the average velocity of all vehicles in the network and a control cost, to penalizes acceleration. The reward function supplied to the learning agent is a weighted combination of the two metrics.

Initialization: The vehicles are evenly spaced around the circular track, with an initial velocity of 0 m/s.

C. Learning setup

The AVs in our system execute controllers which are parameterized control laws, trained using policy gradient methods. For all experiments in this article, we use the *Trust Region Policy Optimization* (TRPO) [78] method for learning the control law, linear feature baselines as described in Duan, et al. [106], discount factor $\gamma = 0.999$, and step size 0.01. Each of the results presented in this article was collected from numerical experiments conducted on three Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz processors for six hours. A total of 6,000,000 samples were collected during the training procedure[3].

D. Results

Through the study of a mixed autonomy single-lane circular track, we demonstrate 1) that Flow enables configuring modules to compose an open traffic control problem and 2) that reliable controllers for complex problems can be efficiently learned, which surpass the performance of all known model-based controllers. This section details our findings, and videos and additional results are available at `https://sites.google.com/view/ieee-tro-flow`.

[3]Further implementation details can be found at: https://github.com/flow-project/flow-lab/tree/master/flow-framework.

1) Performance: Figure 3 shows several key findings. This traffic density versus velocity plot shows the performance of the different learned and model-based controllers. First, we observe that GRU and MLP control laws (in partially observed settings) match the optimal velocity very closely for all trained densities, thereby eliminating congestion. The PI with Saturation and *FollowerStopper* control, on the other hand, only dissipate stop-and-go traffic at densities less than or equal to their calibration density (less congested settings).

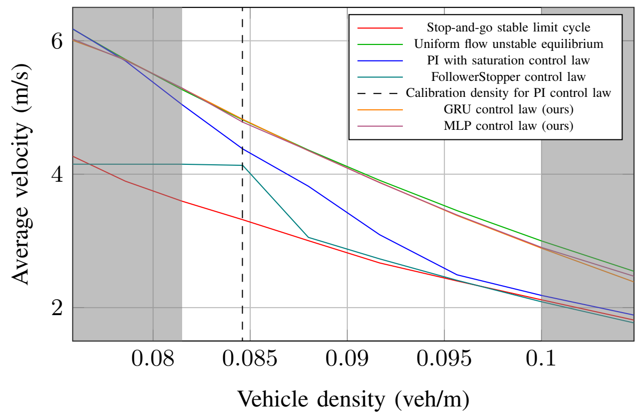

Fig. 3: The performance of the learned (MLP and GRU) and model-based (FollowerStopper and PI Saturation) control laws for various vehicle densities are averaged over ten runs at each evaluated density. Also displayed are the performance upper and lower bounds, derived from the unstable and stable system equilibria, respectively. The GRU and MLP control laws match the velocity upper bound very closely for all densities (train and test). The PI with Saturation and FollowerStopper control laws, on the other hand, only dissipate stop-and-go traffic at densities less than or equal to their calibration density. Remarkably, the GRU and MLP control laws are able to generalize and bring the system to near optimal velocities even at densities outside the training range. Additionally, the learned MLP control law demonstrates that memory is not necessary to achieve near optimal average velocity.

Figure 4 shows the velocity profiles for the different learned and model-based control laws for the 260 m track and additionally includes the *FollowerStopper* control law. We observe that although all controllers are able to stabilize the system, the GRU control law allows the system to reach the uniform flow equilibrium velocity fastest. The GRU and MLP control laws stabilize the system with less oscillatory behavior than the *FollowerStopper* and PI with Saturation control laws, as observed in the velocity profiles. In addition, the *FollowerStopper* control law is the least performant; the control law stabilizes a 260 m track to a speed of 4.15 m/s, well below the 4.82 m/s uniform flow velocity.

Finally, Figure 5 shows the space-time curves for all vehicles in the system, using a variety of control laws. We observe that the PI with Saturation and *FollowerStopper* control law leave much smaller openings in the network (smaller headways) than the MLP and GRU control laws. The MLP control law exhibits the largest openings, as can be seen by the large white portion of the MLP plot within Figure 5. If this were instead applied in a multi-lane circular track, then the smaller openings would have the benefit of preventing opportunistic lane changes, so this observation can lead to better reward design for more complex mixed autonomy traffic studies.

2) Robustness: One of the strengths of the learned GRU and MLP control laws is that they do not rely on external

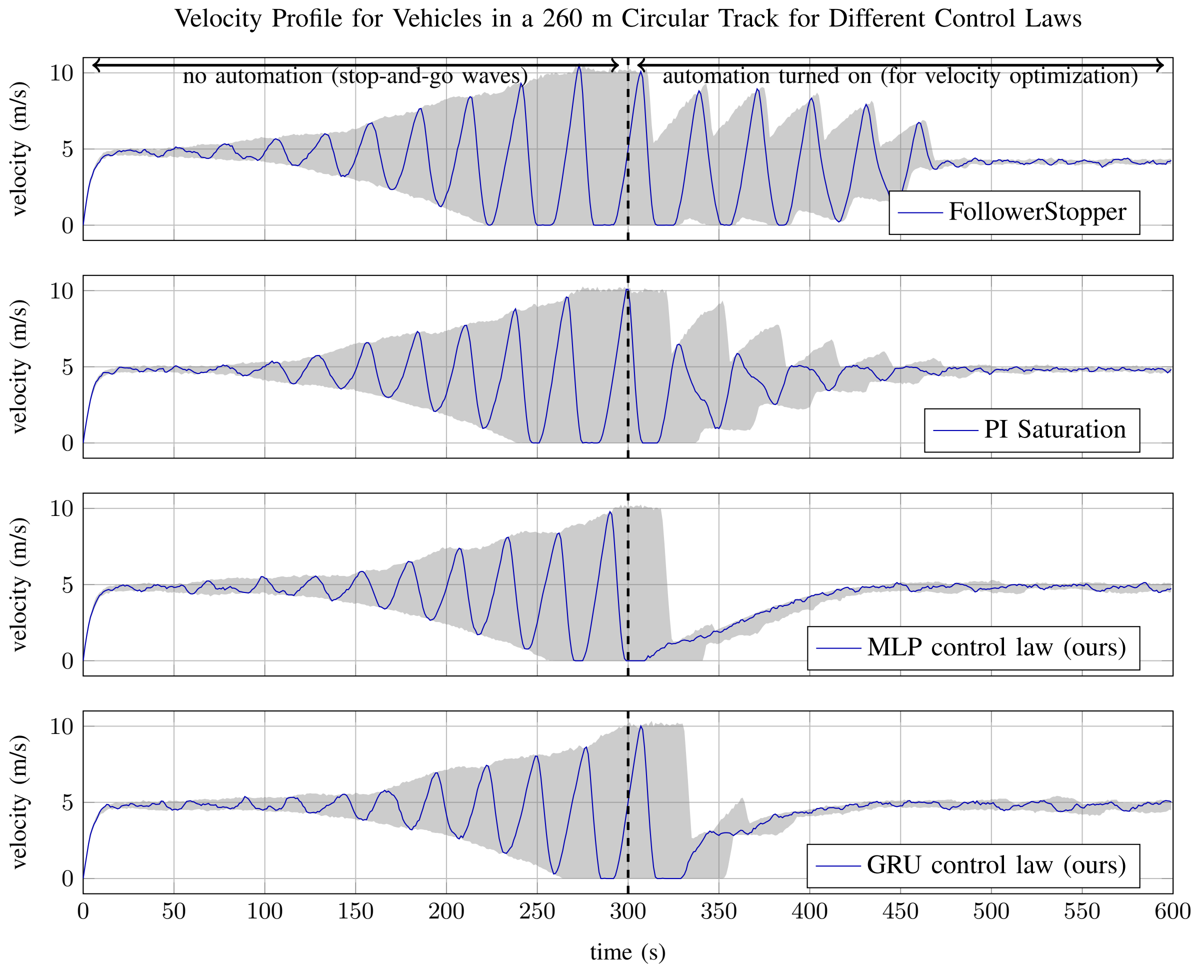

Fig. 4: All experiments start with 300 seconds with the autonomous vehicle following human driving behavior. As we can see from four different control laws, a single autonomous vehicle in a partially observable circular track can stabilize the system. Both learned control laws stabilize the system close to the 4.82 m/s uniform flow velocity, whereas the two model-based controllers fall short. The learned GRU control law allows the system to reach the uniform flow equilibrium velocity fastest.

calibration of parameters that is specific to a particular traffic setting, such as density. Our experience with the model-based controllers, on the other hand, demonstrated considerable sensitivity to the traffic setting. Although the PI with Saturation control law can conceptually adjust to different densities, with its moving average filter, we experimentally found that its performance is sensitive to its parameters (see Figures 3-5). Using parameters calibrated for the 260 m track (as described in Stern, et al. [58]), the PI with Saturation control law performs decently at 260 m; however, this control law's performance quickly degrades at higher densities (more congested settings), dropping close to the performance lower bound.

Relatedly, the *FollowerStopper* control law requires careful tuning before usage, which is beyond the scope of this work. Specifically, the desired velocity must be provided beforehand. Interestingly, we found experimentally that this control law often fails to stabilize the system if provided too high of a desired velocity, even if it is well below the uniform flow equilibrium velocity.

3) Generalization of the learned control law: Training with different vehicle densities encourages the learning of a more robust control law. We found the control law to generalize even to densities outside of the training regime. Figure 3 shows the average velocity of all vehicles in the network achieve for the final 100 s of simulation time; the gray regions indicate the test-time densities. Additionally and surprisingly, when training on different densities but in the *absence* of acceleration error in the human driver models, the learned control laws still successfully stabilized settings *with* human model noise during test time.

4) Partial observability eases controller learning: At this early stage of autonomous vehicle development, we do not yet have a clear picture of what manufacturers will choose in terms of sensing infrastructure for the vehicles, what regulators will require, or what technology will enable (e.g. communication technologies). Furthermore, we do not know how the observation landscape of autonomous vehicles will change over time, as AVs are gradually adopted. Therefore, a methodology which is modular and provides flexibility for the study of AVs is crucial. By invoking the composible observation components, we could readily study a variety of possible scenarios.

This study focuses on the partially-observed setting for several reasons: 1) it is the more realistic setting for near-term deployments of autonomous vehicles, and 2) it permits a fair comparison with previously studied model-based controllers, which typically studied the partially-observed setting. Finally, since our learned control laws already closely track the optimal velocity curve, we do not further explore the fully observed setting. However, in Section VII, we do explore a variety of

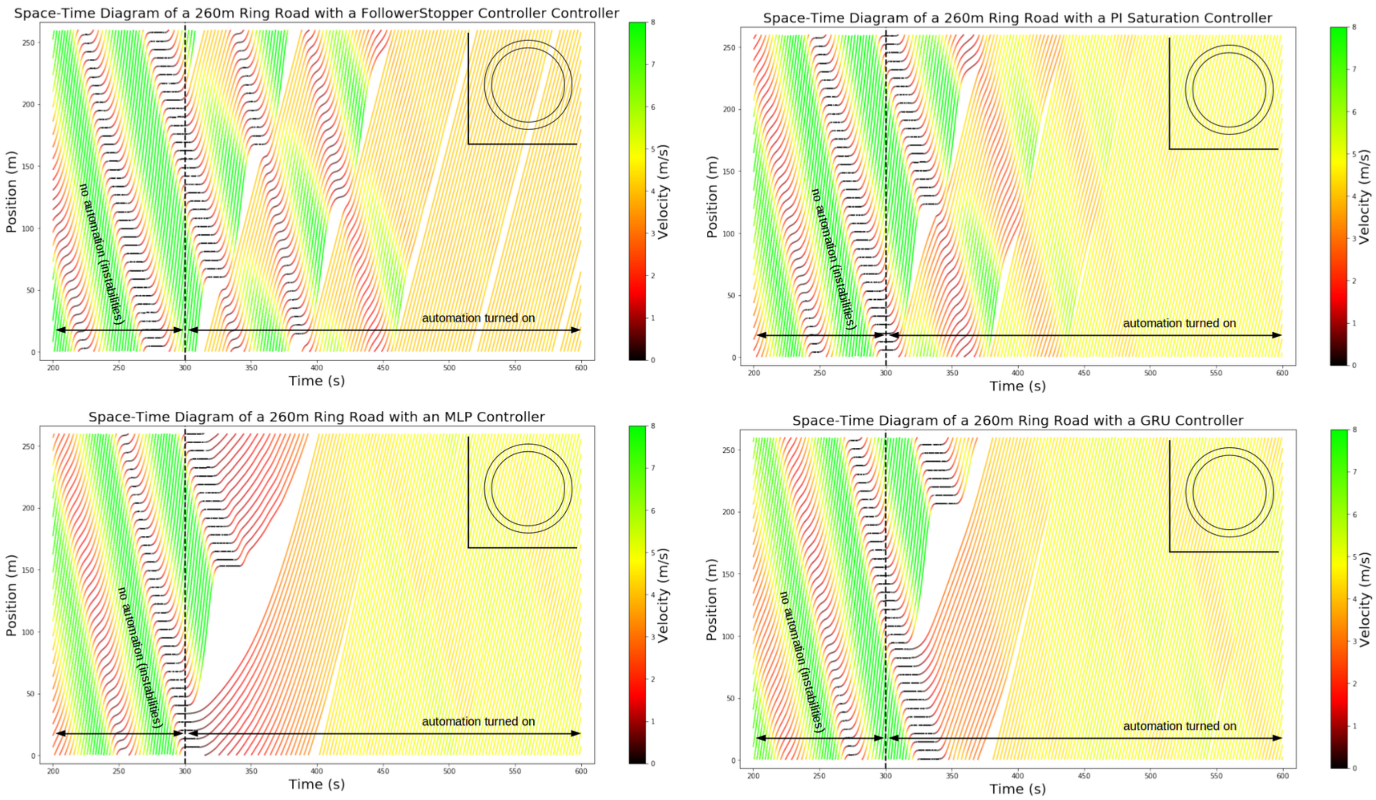

Fig. 5: Prior to the activation of the single AV in the circular track, backward propagating waves are exhibited from stop-and-go traffic resulting from human driving. At 300 seconds, the AV then employs different model-based and learned control laws. **Top left**: Time-space diagram for an AV employing a *FollowerStopper* control law (model-based). **Top right**: Time-space diagram for an AV employing a PI with Saturation control law (model-based). **Bottom left**: Time-space diagram for an AV employing a learned MLP control law. **Bottom right**: Time-space diagram for an AV employing a learned GRU control law.

additional settings, ranging from partially observed to fully observed settings.

Our partially observed experiments uncover several surprising findings, all of which warrant further investigation. First, contrary to the classical view on partially observed control (POMDPs), these experiments suggest that partial observability may ease training instead of making it more difficult; as compared to full observations, we found that partial observations decreased the training time required from around 24 hours to 6 hours. Second, as seen in Figure 3, the results demonstrate that a near global optimum is achievable even under partial observation. Finally, the MLP control law closely mirrors the GRU control law and the optimal velocity curve; despite the partially observed setting, this suggests that memory is not necessary to achieve near optimal velocity across the full range of vehicle densities with a single learned controller.

A possible intuitive explanation is that a neural network with fewer weights may require fewer samples and iterations to converge to a local optimum, thus contributing to faster training. A more rigorous understanding of this phenomenon is left as a topic of future study, as well as questions concerning the situations under which partial observations still lead to a globally optimal solution in a learning framework. These early results suggest that deep RL methods may more efficiently utilize partial observations when they are provided appropriately, avoiding the need to learn to essentially ignore extraneous inputs.

VII. Reusable modules for Mixed Autonomy

The previous section showed that the modules presented in Section V-A could be readily composed to assemble a traffic scenario of interest and to rigorously assess the performance of reinforcement learning methods, regardless of the mathematical framework previously used to assess the scenario. This section goes beyond commonly studied scenarios and demonstrates that the modules can be configured to create new scenarios with important traffic characteristics, such as multiple AVs interacting, lane changes, and intersections, respectively. We present several scenarios to demonstrate the richness of composing simple modules and the insights that can be derived from training controllers for the resulting scenarios. The construction of larger networks with more vehicles or for specific traffic contexts is out of scope of this article, and are additionally limited by the computational efficiency and explainability of current reinforcement learning methods, and are important directions of research.

To demonstrate the ease of the proposed approach, we present only the differences in modules, relative to Section VI. That is, modules not mentioned are plug-and-play from Section VI. Most notably, in contrast to the predominant model-based control approaches for this class of problems,

Network	# vehicles	# AVs	% improvement (avg. speed)
VII-A	22	3	4.07
VII-A	22	11	27.07
VII-B	44	6	6.09
VII-C	14	1	57.45
VII-C	14	14	150.49

TABLE III: AV performance for various experiment configurations. Average velocity improvement is calculated relative to the no-autonomy setting.

limited additional domain knowledge or analysis is required for the study of these more complex traffic control problems. In particular, the reward function, which is a critical piece of any RL experiment, is nearly identical to before: The primary term on system velocity is the same. Note, however, that the specific control cost varies, due to differences in control actions. All methods are compared against a baseline of human performance, as there are limited existing mixed autonomy results as we explore more complex scenarios. Results are summarized in Table III. This set of experiments use a memory-less diagonal Gaussian MLP control law, with hidden layers (100, 50, 25) and `tanh` non-linearity.

A. Single-lane circular track with multiple autonomous vehicles

A simple extension to the previous experiment shows that many variants to the same problem may be studied in a scalable manner, without the need to adhere to strict mathematical frameworks for analysis. The following shows additionally that even simple extensions can yield interesting and significant performance improvements. Here we describe the experimental modules, as they differ from the previous experiment.

Networks: The network is fixed at a circumference of 230 m, as displayed in Figure 1 (top left).

Observer: All vehicle positions and velocities.

Control laws: Three to eleven of the actors are dictated by a single learned control law.

Result: A string of consecutive AVs learns to proceed with a smaller headway than the human driver models (platooning), resulting in greater roadway utilization, thereby permitting a higher velocity for the overall system, as can be seen in Figure 6.

B. Multi-lane circular track with multiple autonomous vehicles

Multi-lane settings are challenging to study from a model-based control theoretic perspective due to the discontinuity in model dynamics from lane changes, as well as due to the complexity of mathematically modeling lane changes accurately. The experimental modules are described here:

Networks: The network is a two-lane circular track of 230 m, as displayed in Figure 1 (top center).

Actors: There are 44 actors, all vehicles.

Observer: All vehicle positions, velocities, and lanes.

Control laws: Six of the actors are dictated by a single learned control law for both acceleration and lane changes

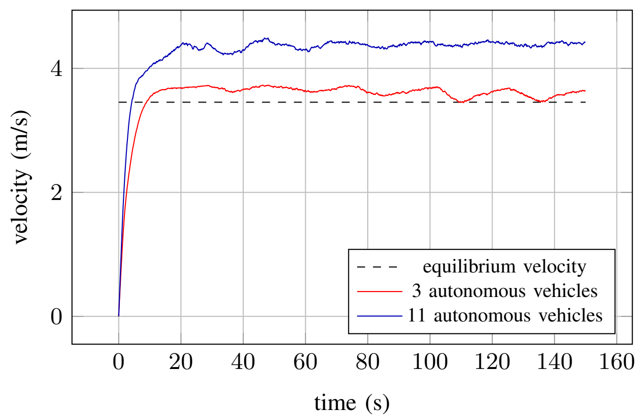

Fig. 6: Velocity profile for single-lane circular track with multiple AVs. With additional AVs, the average velocity exceeds the uniform flow equilibrium velocity, and continues to increases as the number of AVs increase. At three AVs, the average velocity settles at 3.70 m/s; at 11 AVs, the average velocity settles at 4.44 m/s.

(continuous action representation). The rest are dictated by IDM for longitudinal control and the lane changing model of SUMO for lateral control.

Initialization: The vehicles are evenly spaced in the track, using both lanes, and six AVs are placed together in sequence, all in the outer lane.

Result: The learned control law yields AVs balancing across the two lanes, with three in each lane, and avoiding stop-and-go waves in both lanes. The resulting average velocity is 3.66 m/s, an improvement over the 3.45 m/s uniform flow equilibrium velocity. Even though the control inputs are a mix of inherently continuous and discrete signals (acceleration and lane change, respectively), a well-performing control law was learned using only a continuous representation, which is a testament to the flexibility of the approach.

C. Intersection with mixed and full autonomy

We now consider a simplified intersection scenario, which demonstrates the ease of considering different network topologies and traffic rules, such as right-of-way rules at intersections. We additionally demonstrate that the proposed methodology may also be used to study full autonomy settings, in addition to mixed autonomy settings. The authors are not aware of any mixed autonomy control theoretic results for intersections, with which to compare. As a baseline of performance, in the absence of autonomous vehicles, human drivers queue at the intersection, leading to a significant slowdown in the average speed of vehicles in the network.

We now describe the traffic control task in terms of modified modules.

Networks: The network in the shape of a figure eight, as displayed in Figure 1 (top right), with an intersection in the middle, two circular tracks of radius 30 m, and total length of 402 m.

Actors: There are 14 actors, all vehicles.

Observer: All vehicle positions and velocities.

Control laws: One or all of the actors are dictated by a single learned control law for acceleration control inputs. The rest are dictated by IDM for longitudinal control.

Dynamics: There is no traffic light at the intersection; instead vehicles crossing the intersection follow a right-of-way model provided by SUMO to enforce traffic rules and to prevent crashes.

Result: With one autonomous vehicle, the learned control law results in the vehicles moving on average 1.5 times faster. The full autonomy setting exhibits an improvement of almost three times in average velocity. Figure 7 shows the average velocity of vehicles in the network for different levels of autonomy.

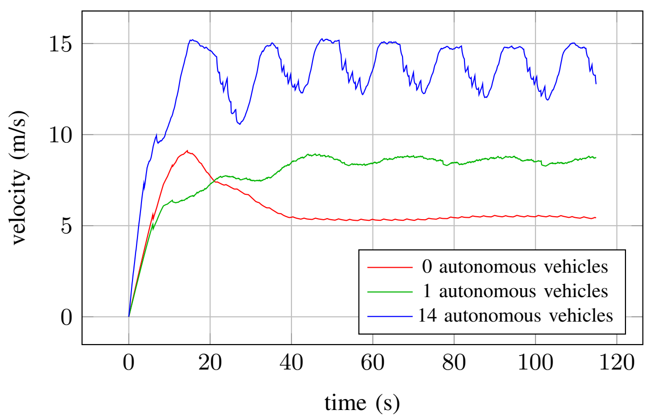

Fig. 7: Velocity profile for the intersection with different fractions of autonomous vehicles. The performance of the network improves as the number of AVs improves; even one autonomous vehicle results in a velocity improvement of 1.5x, and full autonomy almost triples the average velocity from 5 m/s (no autonomy) to 14 m/s.

Despite preliminary work [103], [112], further investigation is needed to understand, interpret, and analyze the learned behaviors and control laws for each of these mixed autonomy experiments, and thereby take steps towards a real-world deployment.

VIII. Conclusion

The complex integration of autonomy into existing systems introduces new technical challenges beyond studying autonomy in isolation or in full. This article aims to make progress on these upcoming challenges by studying how modern deep reinforcement learning can be leveraged to gain insights into complex mixed autonomy systems. In particular, we focus on the integration of autonomous vehicles into urban systems, as we expect these to be among the first of robotic systems to enter and affect existing systems. The article introduces Flow, a modular learning framework which eases the configuration and composition of modules, to enable learning control laws for AVs in complex traffic settings involving nonlinear vehicle dynamics and arbitrary network configurations. Several experiments resulted in controllers which far exceeded state-of-the-art performance (in fact, achieving near-optimal performance) and demonstrated the generality of the methodology for disparate traffic dynamics. Since an early version of this manuscript was made available online in 2017 [95], several works have employed this framework to achieve results in the directions of transfer learning, bottleneck control, sim2real, the discovery of emergent behaviors in traffic, and the design of traffic control benchmarks [112], [113], [114], [103], [115].

Open directions of research include the study of mixed autonomy in larger and more complex traffic networks; studying different, more realistic, and regional objective functions; devising new reinforcement learning techniques for large-scale networked systems; studying scenarios with variable numbers of actors; incorporating advances in safe reinforcement learning; and incorporating multi-agent reinforcement learning for the study of multiple autonomous vehicles. Another open direction is to study fundamental limitations of this methodology; in particular, 1) establishing when guarantees of global convergence, stability, robustness, and safety of classical approaches can not be achieved with deep RL, and 2) quantifying the effects of simulation model error or misspecification on training outcomes. We additionally plan to extend Flow with modules suitable for the study of other forms of automation in traffic, such as problems concerning traffic lights, road directionality, signage, roadway pricing, and infrastructure communication. Another interesting direction is whether an analogous design of reusable modules may be used for other robotics and transportation-related scenarios, such as for motion planning, navigation, ridesharing, and land use planning.

Acknowledgments

The authors would like to thank Professor Alexander Skabardonis for insightful discussions about vehicle dynamics; Leah Dickstein and Nathan Mandi for early prototyping; Nishant Kheterpal, Kathy Jang, Saleh Albeaik and Ananth Kuchibhotla for helping to build out features; Jakob Erdmann for SUMO support; Rocky Duan and Alex Lee for rllab support, used in early experiments; and Philipp Moritz, Eric Liang, and Richard Liaw for Ray and RLlib support. The team is grateful to Professor Dan Work for technical conversations about the circular track experiment, and to the inspirational work of the Piccoli-Seibold-Sprinkle-Work team. The authors would additionally like to thank the anonymous reviewers of early versions of the manuscript for their insightful and constructive comments.

Appendix A
Networks

Flow currently supports learning policies on arbitrary (user-defined) networks. These include closed networks such as single and multi-lane circular tracks, figure-eight networks, and loops with merge as well as open networks, such as intersections, and open networks with such as merge and highway networks with pre-defined in-flows of vehicles into the traffic system. See Figure 1 for various example networks supported by Flow. In each of these networks, Flow can be used to study the design or learning of controllers which optimize the system-level velocity or fuel consumption, in the presence of different types of vehicles, model noise, etc.

Single-lane circular tracks: This network consists of a circular lane with a specified length, inspired by the 230m track studied by Sugiyama et al. [14]. This network has been extensively studied and serves as an experimental and numerical baseline for benchmarking.

Multi-lane circular tracks: Multi-lane circular tracks are a natural extension to problems involving a single circular track. The inclusion of lane-changing behavior in this setting makes studying such problems exceedingly difficult from an analytical perspective, thereby constraining most classical control techniques to the single-lane case. Many multi-lane models forgo longitudinal dynamics in order to encourage tractable analysis [116], [117], [118], [119]. Recent strides have been made in developing simple stochastic models that retain longitudinal dynamics while capturing lane-changing dynamics in a single lane setting [120]. Modern machine learning methods, however, do not require a simplification of the dynamics for the problem to become tractable, as explored in Section VII-B.

Figure-eight network: The figure-eight network is a simple closed network with an intersection. Two circular tracks, placed at opposite ends of the network, are connected by two perpendicular intersections. Vehicles that try to cross this intersection from opposite ends are constrained by a right-of-way model provided by SUMO to prevent crashes.

Loops with merge network: This network permits the study of merging behavior in closed loop networks. This network consists of two circular tracks which are connected together. Vehicles in the smaller track stay within this track, while vehicles in the larger track try to merge into the smaller track and then back out to the larger track. This typically results in congestion at the merge points.

Intersections: This network permits the study of intersection management in an open network. Vehicles arrive in the control zone of the intersection according to a Poisson distribution. At the control zone, the system speeds or slows down vehicles to either maximize average velocity or minimize experienced delay. The building block can be used to build a general schema for arbitrary maps such as the one shown in Figure 1 (bottom right).

APPENDIX B
MODEL-BASED LONGITUDINAL CONTROL LAWS

In this section, we detail the two state-of-the-art control laws for the mixed autonomy circular track, against which we benchmark our learned policies generated using Flow.

1) FollowerStopper: Recent work by [58] presented two control models that may be used by autonomous vehicles to attenuate the emergence of stop-and-go waves in a traffic network. The first of these models is the *FollowerStopper*. This model commands the AVs to maintain a desired velocity U, while ensuring that the vehicle does not crash into the vehicle behind it. Following this model, the command velocity v^{cmd} of the autonomous vehicle is:

$$v^{\text{cmd}} = \begin{cases} 0 & \text{if } \Delta x \leq \Delta x_1 \\ v\frac{\Delta x - \Delta x_1}{\Delta x_2 - \Delta x_1} & \text{if } \Delta x_1 < \Delta x \leq \Delta x_2 \\ v + (U - v)\frac{\Delta x - \Delta x_2}{\Delta x_3 - \Delta x_2} & \text{if } \Delta x_2 < \Delta x \leq \Delta x_3 \\ U & \text{if } \Delta x_3 < \Delta x \end{cases} \tag{5}$$

where $v = \min(\max(v^{\text{lead}}, 0), U)$, v^{lead} is the speed of the leading vehicles, Δx is the headway of the autonomous vehicle, subject to boundaries defined as:

$$\Delta x_k = \Delta x_k^0 + \frac{1}{2d_k}(\Delta v_-)^2, \quad k = 1, 2, 3 \tag{6}$$

The parameters of this model can be found in [58].

2) PI with Saturation: In addition to the *FollowerStopper* control law, [58] presents a model called the *PI with Saturation control law* that attempts to estimate the average equilibrium velocity U for vehicles on the network, and then drives at that speed. This average is computed as a temporal average from its own history: $U = \frac{1}{m}\sum_{j=1}^{m} v_j^{AV}$. The target velocity at any given time is then defined as:

$$v^{\text{target}} = U + v^{\text{catch}} \times \min\left(\max\left(\frac{\Delta x - g_l}{g_u - g_l}, 0\right), 1\right) \tag{7}$$

Finally, the command velocity for the vehicle at time $j+1$, which also ensures that the vehicle does not crash, is:

$$v_{j+1}^{\text{cmd}} = \beta_j(\alpha_j v_j^{\text{target}} + (1-\alpha_j)v_j^{\text{lead}}) + (1-\beta_j)v_j^{\text{cmd}} \tag{8}$$

The values for all parameters in the model can be found in [58] and are also provided in Table I.

REFERENCES

[1] Z. Wadud, D. MacKenzie, and P. Leiby, "Help or hindrance? the travel, energy and carbon impacts of highly automated vehicles," *Transportation Research Part A: Policy and Practice*, vol. 86, pp. 1–18, 2016.

[2] U. DOT, "National transportation statistics," *Bureau of Transportation Statistics, Washington, DC*, 2016.

[3] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[4] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*. Springer, 2009, vol. 56.

[5] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore," in *Road Vehicle Automation*. Springer, 2014, pp. 229–245.

[6] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[7] B. K. Horn, "Suppressing traffic flow instabilities," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 13–20.

[8] D. Miculescu and S. Karaman, "Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals," *arXiv preprint arXiv:1607.07896*, 2016.

[9] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[10] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *CoRR*, vol. abs/1610.03295, 2016. [Online]. Available: http://arxiv.org/abs/1610.03295

[11] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *ICML*, vol. 97. Citeseer, 1997, pp. 12–20.

[12] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[13] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

	PI with Saturation			
Parameters	δ	g_l	g_u	v^{catch}
Values	2 m	7 m	30 m	1 m/s

	FollowerStopper										
Parameters	Δx_1^0	Δx_2^0	Δx_3^0	d_1	d_2	d_3	Δv	Δx_1	Δx_2	Δx_3	U
Values	4.5 m	5.25 m	6.0 m	1.5 m/s^2	1.0 m/s^2	0.5 m/s^2	-3 m/s	7.5 m/s	9.75 m/s	15 m/s	4.15 m/s

TABLE IV: Parameters for model-based controllers

[14] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks–experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, p. 033001, 2008.

[15] G. Orosz, R. E. Wilson, and G. Stépán, "Traffic jams: dynamics and control," *Philosophical Trans. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4455–4479, 2010.

[16] M. J. Lighthill and G. B. Whitham, "On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 229, pp. 317–345, 1955.

[17] P. I. Richards, "Shock Waves on the Highway," *Operations Research*, vol. 4, pp. 42–51, 1956.

[18] H. J. Payne, "Freflo: A macroscopic simulation model of freeway traffic," *Transportation Research Record*, no. 722, 1979.

[19] A. Aw and M. Rascle, "Resurrection of" second order" models of traffic flow," *SIAM journal on applied mathematics*, vol. 60, no. 3, pp. 916–938, 2000.

[20] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[21] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[22] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Structure stability of congestion in traffic dynamics," *Japan Journal of Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 203–223, 1994.

[23] ——, "Dynamical model of traffic congestion and numerical simulation," *Physical review E*, vol. 51, no. 2, p. 1035, 1995.

[24] A. J. Miller, "A queueing model for road traffic flow," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 23, no. 1, pp. 64–75, 1961.

[25] D. Heidemann, "A queueing theory approach to speed-flow-density relationships," in *TRANSPORTATION AND TRAFFIC THEORY. PROCEEDINGS OF THE 13TH INTERNATIONAL SYMPOSIUM ON TRANSPORTATION AND TRAFFIC THEORY, LYON, FRANCE, 24-26 JULY 1996*, 1996.

[26] T. Van Woensel and N. Vandaele, "Modeling traffic flows with queueing models: a review," *Asia-Pacific Journal of Operational Research*, vol. 24, no. 04, pp. 435–461, 2007.

[27] S. Thrun, W. Burgard, D. Fox *et al.*, *Probabilistic robotics*. MIT press Cambridge, 2005, vol. 1.

[28] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572–578, Jun 1999.

[29] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, Jun 2001.

[30] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with gps in urban canyon environments," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 15–25, Feb 2003.

[31] Z. Shiller and Y. R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, Apr 1991.

[32] S. D. Bopardikar, B. Englot, and A. Speranzon, "Multiobjective path planning: Localization constraints and collision probability," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 562–577, June 2015.

[33] J. Minguez and L. Montano, "Extending collision avoidance methods to consider the vehicle shape, kinematics, and dynamics of a mobile robot," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 367–381, April 2009.

[34] K. Kanatani and K. Watanabe, "Reconstruction of 3-d road geometry from images for autonomous land vehicles," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 127–132, Feb 1990.

[35] S. Drakunov, U. Ozguner, P. Dix, and B. Ashrafi, "Abs control using optimum search via sliding modes," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 79–85, March 1995.

[36] B. Van Arem, C. J. Van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, 2006.

[37] J. Lee, J. Choi, K. Yi, M. Shin, and B. Ko, "Lane-keeping assistance control algorithm using differential braking to prevent unintended lane departures," *Control Engineering Practice*, vol. 23, pp. 1–13, 2014.

[38] Y. S. Son, W. Kim, S.-H. Lee, and C. C. Chung, "Robust multirate control scheme with predictive virtual lanes for lane-keeping system of autonomous highway driving," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3378–3391, 2015.

[39] S. Lefevre, Y. Gao, D. Vasquez, H. E. Tseng, R. Bajcsy, and F. Borrelli, "Lane keeping assistance with learning-based driver model and model predictive control," in *12th International Symposium on Advanced Vehicle Control*, 2014.

[40] C. Hatipoglu, U. Ozguner, and K. A. Redmill, "Automated lane change controller design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 1, pp. 13–22, March 2003.

[41] B. Corporation, "Four wheels on jacks park car," *Popular Science Monthly*, September 1934. [Online]. Available: https://books.google.com/?id=HCgDAAAAMBAJ&pg=PA58&dq=Popular+Science+1931+plane#v=onepage&q&f=true

[42] I. E. Paromtchik and C. Laugier, "Motion generation and control for parking an autonomous vehicle," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 4. IEEE, 1996, pp. 3117–3122.

[43] V. Milans, D. F. Llorca, J. Villagr, J. Prez, C. Fernndez, I. Parra, C. Gonzlez, and M. A. Sotelo, "Intelligent automatic overtaking system using vision for vehicle detection," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3362 – 3373, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411013339

[44] D. Miculescu and S. Karaman, "Polling-systems-based control of high-performance provably-safe autonomous intersections," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 1417–1423.

[45] Technical Committee ISO/TC 204, Intelligent transport systems, *Intelligent transport systems – Adaptive Cruise Control systems – Performance requirements and test procedures*, ISO ISO 15 622:2010, 2010.

[46] P. A. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. on Vehicular technology*, vol. 42, no. 4, pp. 657–672, 1993.

[47] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE transactions on intelligent transportation systems*, vol. 4, no. 3, pp. 143–153, 2003.

[48] S. E. Shladover, "Automated vehicles for highway operations (automated highway systems)," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 219, no. 1, pp. 53–75, 2005.

[49] R. Rajamani and C. Zhu, "Semi-autonomous adaptive cruise control systems," *IEEE Trans. on Vehicular Technology*, vol. 51, no. 5, pp. 1186–1192, 2002.

[50] X. Lu, S. Shladover, and J. Hedrick, "Heavy-duty truck control: Short inter-vehicle distance following," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 5. IEEE, 2004, pp. 4722–4727.

[51] S. Sheikholeslam and C. A. Desoer, "A system level study of the longitudinal control of a platoon of vehicles," *Journal of dynamic systems, measurement, and control*, vol. 114, no. 2, pp. 286–292, 1992.

[52] S. Cui, B. Seibold, R. Stern, and D. B. Work, "Stabilizing traffic flow via a single autonomous vehicle: Possibilities and limitations,"

in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 447–453.

[53] C. Wu, A. M. Bayen, and A. Mehta, "Stabilizing traffic with autonomous vehicles," in *Submission to IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[54] A. Kesting *et al.*, "Jam-avoiding adaptive cruise control (ACC) and its impact on traffic dynamics," in *Traffic and Granular Flow05*. Springer, 2007, pp. 633–643.

[55] Y.-M. Yuan, R. Jiang, M.-B. Hu, Q.-S. Wu, and R. Wang, "Traffic flow characteristics in a mixed traffic system consisting of acc vehicles and manual vehicles: A hybrid modelling approach," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 12, pp. 2483–2491, 2009.

[56] T.-C. Au, S. Zhang, and P. Stone, "Semi-autonomous intersection management," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1451–1452.

[57] G. Sharon and P. Stone, "A protocol for mixed autonomous and human-operated vehicles at intersections," in *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 2017, pp. 151–167.

[58] R. E. Stern, S. Cui, M. L. D. Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, R. Haulcy, H. Pohlmann, F. Wu, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *CoRR*, vol. abs/1705.01693, 2017. [Online]. Available: http://arxiv.org/abs/1705.01693

[59] M. Garavello and B. Piccolli, *Traffic flow on networks: Conservation Laws Models*. Springfield, MO: American Institute of Mathematical Sciences, 2006.

[60] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012.

[61] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: a queueing-theoretical perspective," 2014.

[62] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and Systems*, 2016.

[63] J. Lee, M. Park, and H. Yeo, "A probability model for discretionary lane changes in highways," *KSCE Journal of Civil Engineering*, vol. 20, no. 7, pp. 2938–2946, 2016.

[64] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[65] ——, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2017.

[66] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[67] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[68] C.-Y. Liang and P. Huei, "String stability analysis of adaptive cruise controlled vehicles," *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, vol. 43, no. 3, pp. 671–677, 2000.

[69] A. Bose and P. A. Ioannou, "Analysis of traffic flow with mixed manual and semiautomated vehicles," *IEEE Trans. on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 173–188, 2003.

[70] P. A. Ioannou and M. Stefanovic, "Evaluation of acc vehicles in mixed traffic: Lane change effects and sensitivity analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 79–89, 2005.

[71] M. A. S. Kamal, J.-i. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "Smart driving of a vehicle using model predictive control for improving traffic flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 878–888, 2014.

[72] D. Swaroop, "String stability of interconnected systems: An application to platooning in automated highway systems," *California Partners for Advanced Transit and Highways (PATH)*, 1997.

[73] G. Orosz, J. Moehlis, and F. Bullo, "Delayed car-following dynamics for human and robotic drivers," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2011, pp. 529–538.

[74] I. G. Jin and G. Orosz, "Dynamics of connected vehicle systems with delayed acceleration feedback," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 46–64, 2014.

[75] L. Wang, B. K. Horn, and G. Strang, "Eigenvalue and eigenvector analysis of stability for a line of traffic," *Studies in Applied Mathematics*, 2016.

[76] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[77] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[78] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust region policy optimization," in *ICML*, 2015, pp. 1889–1897.

[79] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa, "Learning continuous control policies by stochastic value gradients," in *Advances in Neural Information Processing Systems*, 2015, pp. 2944–2952.

[80] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *J. Artif. Intell. Res.(JAIR)*, vol. 47, pp. 253–279, 2013.

[81] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik *et al.*, "Deepmind lab," *arXiv preprint arXiv:1612.03801*, 2016.

[82] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[83] G. Synnaeve, N. Nardelli, A. Auvolat, S. Chintala, T. Lacroix, Z. Lin, F. Richoux, and N. Usunier, "Torchcraft: a library for machine learning research on real-time strategy games," *arXiv preprint arXiv:1611.00625*, 2016.

[84] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5026–5033.

[85] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, 2000.

[86] M.-A. Côté, Á. Kádár, X. Yuan, B. Kybartas, T. Barnes, E. Fine, J. Moore, M. Hausknecht, L. E. Asri, M. Adada *et al.*, "Textworld: A learning environment for text-based games," *arXiv preprint arXiv:1806.11532*, 2018.

[87] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[88] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[89] L. Li, Y. Lv, and F. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, July 2016.

[90] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1198–1207, 2018.

[91] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," *arXiv preprint arXiv:1905.05717*, 2019.

[92] H. Wei, G. Zheng, H. Yao, and Z. 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*. ACM, 2018, pp. 2496–2505.

[93] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[94] C. Wu, K. Parvate, N. Kheterpal, L. Dickstein, A. Mehta, E. Vinitsky, and A. M. Bayen, "Framework for control and deep reinforcement learning in traffic," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, 2017.

[95] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv preprint arXiv:1710.05465*, 2017.

[96] R. Bellman, "A markovian decision process," DTIC Document, Tech. Rep., 1957.

[97] R. A. Howard, "Dynamic programming and markov processes," 1964.

[98] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.
[99] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
[100] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International Conference on Machine Learning*, 2015, pp. 2067–2075.
[101] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.
[102] M. Treiber, A. Kesting, and D. Helbing, "Multi-anticipative driving in microscopic traffic models," Tech. Rep., 2004.
[103] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Emergent behaviors in mixed-autonomy traffic," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 398–407. [Online]. Available: http://proceedings.mlr.press/v78/wu17a.html
[104] Y. Zheng, J. Wang, and K. Li, "Smoothing traffic flow via control of autonomous vehicles," *arXiv preprint arXiv:1812.09544*, 2018.
[105] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.
[106] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," *CoRR*, vol. abs/1604.06778, 2016. [Online]. Available: http://arxiv.org/abs/1604.06778
[107] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, "Ray rllib: A composable and scalable reinforcement learning library," *arXiv preprint arXiv:1712.09381*, 2017.
[108] C. Wu, A. M. Bayen, and A. Mehta, "Stabilizing traffic with autonomous vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.
[109] CIECA, "Internal project on eco-driving in category b driver training & the driving test," 2007.
[110] M. Barth and K. Boriboonsomsin, "Energy and emissions impacts of a freeway-based dynamic eco-driving system," *Transportation Research Part D: Transport and Environment*, vol. 14, no. 6, pp. 400 – 410, 2009, the interaction of environmental and traffic safety policies. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361920909000121
[111] M. Treiber and A. Kesting, "The intelligent driver model with stochasticity-new insights into traffic flow oscillations," *Transportation Research Procedia*, vol. 23, pp. 174–187, 2017.
[112] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1475–1480.
[113] E. Vinitsky, K. Parvate, A. Kreidieh, C. Wu, and A. Bayen, "Lagrangian control through deep-rl: Applications to bottleneck decongestion," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 759–765.
[114] K. Jang, E. Vinitsky, B. Chalaki, B. Remer, L. Beaver, A. A. Malikopoulos, and A. Bayen, "Simulation to scaled city: zero-shot policy transfer for traffic control via autonomous vehicles," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. ACM, 2019, pp. 291–300.
[115] E. Vinitsky, A. Kreidieh, L. Le Flem, N. Kheterpal, K. Jang, F. Wu, R. Liaw, E. Liang, and A. M. Bayen, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Conference on Robot Learning*, 2018, pp. 399–409.
[116] P. G. Michalopoulos, D. E. Beskos, and Y. Yamauchi, "Multilane traffic flow dynamics: some macroscopic considerations," *Transportation Research Part B: Methodological*, vol. 18, no. 4, pp. 377–395, 1984.
[117] A. Klar and R. Wegener, "A hierarchy of models for multilane vehicular traffic i: Modeling," *SIAM Journal on Applied Mathematics*, vol. 59, no. 3, pp. 983–1001, 1998.
[118] A. Sasoh and T. Ohara, "Shock wave relation containing lane change source term for two-lane traffic flow," *Journal of the Physical Society of Japan*, vol. 71, no. 9, pp. 2339–2347, 2002.
[119] C. F. Daganzo, "A behavioral theory of multi-lane traffic flow. part i: Long homogeneous freeway sections," *Transportation Research Part B: Methodological*, vol. 36, no. 2, pp. 131–158, 2002.
[120] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Multi-lane reduction: A stochastic single-lane model for lane changing," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, 2017.