Unified Automatic Control of Vehicular Systems With Reinforcement Learning

Zhongxia Yan, *Member, IEEE*, Abdul Rahman Kreidieh, *Member, IEEE*, Eugene Vinitsky, *Member, IEEE*, Alexandre M. Bayen, *Senior Member, IEEE*, and Cathy Wu, *Member, IEEE*

***Abstract*— Emerging vehicular systems with increasing proportions of automated components present opportunities for optimal control to mitigate congestion and increase efficiency. There has been a recent interest in applying deep reinforcement learning (DRL) to these nonlinear dynamical systems for the automatic design of effective control strategies. Despite conceptual advantages of DRL being model-free, studies typically nonetheless rely on training setups that are painstakingly specialized to specific vehicular systems. This is a key challenge to efficient analysis of diverse vehicular and mobility systems. To this end, this article contributes a streamlined methodology for vehicular microsimulation and discovers high performance control strategies with minimal manual design. A variable-agent, multi-task approach is presented for optimization of vehicular Partially Observed Markov Decision Processes. The methodology is experimentally validated on mixed autonomy traffic systems, where fractions of vehicles are automated; empirical improvement, typically 15-60% over a human driving baseline, is observed in all configurations of six diverse open or closed traffic systems. The study reveals numerous emergent behaviors resembling wave mitigation, traffic signaling, and ramp metering. Finally, the emergent behaviors are analyzed to produce interpretable control strategies, which are validated against the learned control strategies.**

***Note to Practitioners*—As vehicular systems such as real-world traffic systems and robotic warehouses become increasingly automated, optimizing vehicle movements sees an increasing potential to reduce congestion and increase efficiency. For many vehicular systems, simulations of varying fidelity are commonly used for analysis and optimization without the need to deploy real vehicles. This article describes a unified and practical approach for optimal control of vehicles in arbitrary simulated vehicular systems while permitting partial automation, where the behavior of fractions of vehicles at given times can be modelled but not controlled. As illustrated by the diverse traffic systems considered in this article, the presented methodology emphasizes ease of application within any simulated vehicular system while minimizing manual efforts by the practitioner. The control inputs consist of local information around each automated vehicle, while the control outputs are commands for longitudinal acceleration and lateral lane change. Experimental results are presented for relatively small simulated traffic systems, though the methodology can be adapted to larger vehicular systems with minor modifications. Experimentally optimized behaviors provide insights to the practitioner which may assist in designing simplified and interpretable control strategies. Implementation in real-world systems depends on two requirements: 1) a reliable fallback mechanism for ensuring safety of vehicles, and 2) sufficient fidelity of the simulator for simulated behaviors to transfer. These requirements are under active research for traffic systems and may be practical in some robotic settings. To facilitate robust transfer of policies from simulated to real-world systems, future extensions of this work may inject additional randomization into simulation while reducing the unmodeled stochasticity of targeted real-world systems as much as possible.**

***Index Terms*— Primary topics: Mobile traffic control, automated vehicles, reinforcement learning. Secondary topic keywords: Mixed autonomy, multi-agent systems.**

Manuscript received January 7, 2022; revised March 10, 2022; accepted April 16, 2022. This article was recommended for publication by Associate Editor M. Robba and Editor J. Yi upon evaluation of the reviewers' comments. This work was supported in part by Amazon, in part by the MIT-IBM Watson AI Laboratory, and in part by the Department of Transportation Dwight David Eisenhower Transportation Fellowship Program. *(Corresponding author: Zhongxia Yan.)*

Zhongxia Yan is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: zxyan@mit.edu).

Abdul Rahman Kreidieh is with the Department of Civil and Environmental Engineering, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: aboudy@berkeley.edu).

Eugene Vinitsky is with the Department of Mechanical Engineering, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail:evinitsky@berkeley.edu).

Alexandre M. Bayen is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: bayen@berkeley.edu).

Cathy Wu is with the Laboratory for Information and Decision Systems, the Department of Civil and Environmental Engineering, and the Institute of Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: cathywu@mit.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3168621.

Digital Object Identifier 10.1109/TASE.2022.3168621

I. Introduction

A DEVELOPING trend in mobility systems today is the full or partial adoption of automated control of mobile vehicles in traditionally human-operated roles [1]. This trend can be observed in systems ranging from real-world traffic systems [2] to warehouses employing mobile robots for storage, sorting, or delivery [3]. Increasing autonomy in these systems increases the potential to algorithmically control and coordinate automated vehicles (AVs) to increase efficiency, reduce congestion, or optimize other objectives like fuel usage throughout the system.

For the near future, while AV adoption remains fractional, automated control in real-world traffic systems would necessarily interact with human control, creating *mixed autonomy* traffic. While any control may be underactuated in the mixed autonomy setting, such control may still induce desired behaviors, as demonstrated by several recent works on reducing congestion in simulated, mixed autonomy circular [4] or highway traffic systems [5].

In many cases, mixed or full automation must solve an underlying mixed discrete and continuous optimization problem, which may be difficult to even formulate due to complex and stochastic dynamics, let alone solve practically. For many such complex systems, simulation decouples modeling of the

system from further analysis and optimization efforts. For this reason, researchers and practitioners construct simulations of varying fidelity for many real-world systems.

In this study, we demonstrate the generality and ease of applicability of a unified model-free deep reinforcement learning (DRL)-based methodology for optimizing behaviors in diverse mixed autonomy traffic systems in simulation. In contrast to planning and search algorithms, model-free reinforcement learning is applicable to both continuous and discrete domains and only requires the ability to simulate forward trajectories from a set of initial states; moreover, learned DRL policies may execute more efficiently in real-time and at scale.

We acknowledge that a sizeable gap exists between simulation and reality, especially in real-world traffic systems with many stochastic actors with human-intent. While preliminary Sim2Real policy transfer has been done in miniature physical traffic systems [6], we do not expect Sim2Real policy transfer to be feasible for real-world traffic systems in the near future [7]. Nevertheless, we attempt to derive insights and interpretable controllers from the learned policies. Toward this end, previous works have designed simulated automation and control strategies for industrial warehouse vehicles [8], metro train regulation [9], airport surface management [10], container loading [11], and even pedestrian control [12]. Moreover, we argue that near-future Sim2Real extensions of our work *are feasible* for fully automated robotic systems, which may require movement and coordination of automated vehicular robots with assigned routes [3]. These settings are suitable due to 1) existence of higher fidelity simulators, 2) little or no need to simulate human intent, and 3) existence of collision avoidance mechanisms.

This work follows a series of our previous works applying DRL to mixed autonomy traffic [4], [13]–[17]. While each previous work often focuses on a single traffic system and applies significant amounts of system-specific handcrafting (indeed, DRL-based methods in various applications are known to be notoriously hard to tune to good performance), this work presents a simplified and unified DRL methodology for a superset of open and closed traffic systems, with a focus on ease of applicability. The code introduced in this work is a lightweight revision of the Flow Framework [4], completely rewritten to offer researchers and practitioners more control in designing the traffic system while minimizing the amount of DRL design choices and hyperparameters. Additionally, we interpret the behaviors of DRL-controlled AVs, some of which resemble those designed by traffic engineering experts, and design simple controllers inspired by the learned DRL policies. While our previous works may offer deeper insights into the performance of DRL in particular traffic systems, this work emphasizes the ease of applicability of model-free DRL to general vehicular systems with mixed or full autonomy.

In summary, the contributions of our present work are:

1) We present a unified variable-agent, multi-task DRL methodology and showcase the generality, effectiveness, and ease of usage for optimizing mixed autonomy traffic in simulated vehicular systems.
2) To shed light on the performant behaviors discovered automatically via DRL, we manually extract and benchmark simple controllers inspired by the behaviors.
3) We characterize the robustness of each trained policy across a range of vehicle densities.
4) We introduce a lightweight codebase with heavy emphasis on ease of usage and simplified design choices for researchers and practitioners.

Code, models, and videos of results are available on Github.

The rest of this article is organized as such: Section II details related work, Section III introduces relevant DRL concepts, Section IV defines relevant vehicular systems, Section V details the DRL methodology, Section VI discusses experimental setup, and Section VII provides experimental results.

II. Related Work

A. Traffic Control

Due to the ubiquity and costs of congestion in traffic, much work has been devoted to traffic control for reducing congestion and increasing efficiency [18]. In urban traffic networks, composed of many intersections, traffic signal control strategies have been widely studied and sometimes deployed for isolated or coordinated intersections, including fixed-time [19] or adaptive [20]. In freeway traffic networks [21], ramp metering control methods like ALINEA [22] are deployed to counter congestion due to reduction in road capacity, and cooperative adaptive cruise control (CACC) [23] methods are proposed to mitigate congestion due to perturbations in traffic flow. While works in CACC and our study both concern vehicular control of traffic, we aim to automatically discover optimal traffic behavior rather than manually prescribing desired speeds for vehicles to follow.

B. Isolated Autonomy

The US DARPA challenges in autonomous driving spurred much research in components necessary for the deployment of automated vehicles in the real-world [24]. These developments, along with developments in advanced driver assistance systems (ADAS) like adaptive cruise control (ACC) [25], focus on the safety, comfort, "human-ness," and performance of an individually automated vehicle rather than the traffic system as a whole [26].

C. Mixed and Full Autonomy

Unlike isolated autonomy, mixed and full autonomy are often studied as traffic control techniques aimed to optimize local or system-wide objectives. CACC methods [23] are often studied under mixed and full autonomy settings with varying penetration rates of CACC vehicles. Mixed and full autonomy control of intersections have been studied by [27] and [28]–[30], respectively. However, the former analyzes the performance of a fixed first-come-first-serve protocol while the latter abstracts away intersection dynamics into a polling-system of two queues. As discussed in more detail in [4], two prominent challenges in studying mixed autonomy in particular are the high uncertainty in system dynamics, due

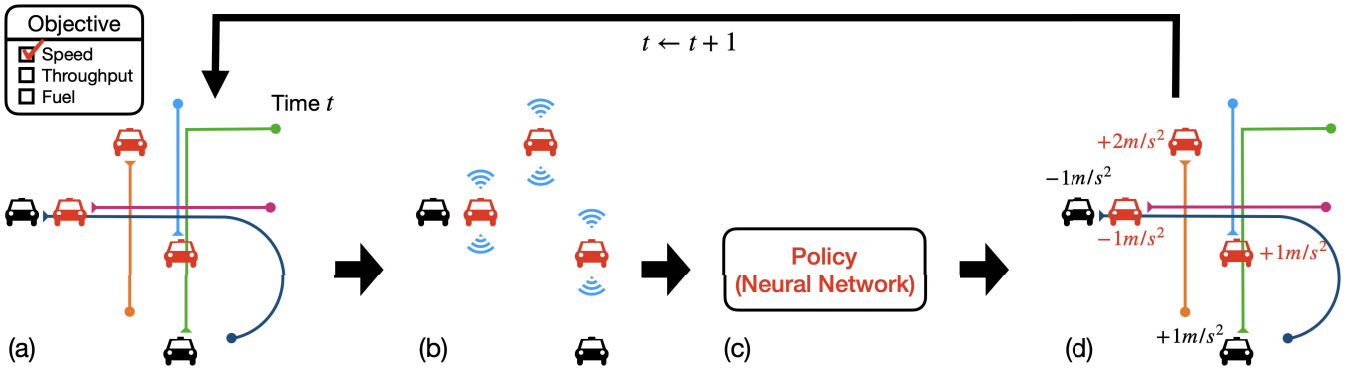


Fig. 1. **An exemplary vehicular system with an overview of our methodology.** a) At each time t, all vehicles have assigned routes towards assigned destinations; a fraction of vehicles are automated (red) while the rest are uncontrolled (black). We would like to control the AVs towards maximizing a desired objective. b) We define the set of neighboring vehicles that each AV may sense. c) Based on each automated vehicle's observed surrounding, a learned control policy dictates the action of each AV towards optimization of the objective. Uncontrolled vehicles follow some default policy. d) Position and speeds of all vehicles are updated, and the process repeats at time $t+1$.

to modeling human behavior, and the lack of a known optimal behavior. As we show in this work and our previous works, DRL may be a suitable methodology which addresses both challenges. Throughout this work, we assume that all control decisions are supported by ideal vehicular communication, and we defer to [31], [32] for a discussions realistic, non-ideal communication in vehicular systems.

D. Model-Free Reinforcement Learning

Derived from optimal control and machine learning, model-free reinforcement learning (RL) is a methodology for optimizing sequential decision making [33], [34]. Model-free RL decouples optimal control from system modeling with Markov Decision Process as the interface. Addressing the challenges of mixed autonomy, model-free RL does not need to model the dynamics of the underlying system and does not require knowledge of an optimal behavior. Recently, model-free DRL, combining model-free RL with deep neural networks, has demonstrated improved performance for traffic signal control [35], ramp metering [36], and multi-robot navigation [37]. Other applications of model-free DRL to automation include optimal parameters for computer numerical control (CNC) machining [38] and optimal scheduling in manufacturing [39], [40]. However, as model-free DRL algorithms are primarily simulation-based, deployment in real-world settings suffers from several difficulties [7]; in this article, we briefly acknowledge the gap between simulation and deployment.

E. Model-Free DRL for Mixed Autonomy Traffic

This work generalizes our previous works on applications of model-free DRL to mixed autonomy traffic systems based on the Flow framework [4], [5], [13]–[17], [41]. While each previous work demonstrates that DRL overcomes long-standing classical control challenges in traffic control, including complex dynamics models, long horizons, partial observations, and non-standard noise, these work often included artificial encouragement and handcrafting to guide the DRL policy in their specific traffic system.

For reward shaping, [41] and [4] penalize acceleration and deceleration to encourage convergence to a constant speed in ring-like traffic systems while [13]–[15] penalize deviation from tuned desired speed hyperparameters in figure-eight and highway ramp traffic systems. On the other hand, [5] and [16] restricts control over AVs to selected segments of a highway bottleneck system to encourage ramp metering-like behavior. In both cases, reward shaping and selective control of vehicles not only require cumbersome tuning the researcher or practitioner, but are constrained by human intuition which may be suboptimal in more complex systems. Moreover, [4], [5], [13]–[16], [41] all use actor-critic algorithms ranging from TRPO [42] to TD3 [43] which require extensive hyperparameter tuning for training neural networks for both the policy and a value function critic; in contrast, this work shows that the TRPO algorithm with only a policy neural network and without a value function network suffices for all considered traffic systems, eliminating much of the hyperparameter tuning.

Overall, this work shows that a unified DRL methodology achieves similar or better efficiency without resorting to system-specific hand-designing to ease optimization. We believe that the ability to easily discover performant behaviors in any system without hand-holding is key towards broader applicability of DRL in general vehicular systems.

III. Preliminaries

A. Markov Decision Process (MDP)

Markov Decision Process (MDP) is a framework for modeling sequential decisions. We model each decision process in this paper as a finite-horizon discounted MDP, defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \rho_0, \gamma, H)$ consisting of state space $\mathcal{S}$, action space $\mathcal{A}$, stochastic transition function $T(s, a, s') = p(s'|s, a)$ for $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$, reward function $r(s, a, s') \in \mathbb{R}$, initial state distribution ρ_0, discount factor $\gamma \in [0, 1]$, and horizon $H \in \mathbb{Z}_+$. Given this MDP definition, reinforcement learning and optimal control aim towards the following objective

$$\max_{a_0 \ldots a_{H-1} \in \mathcal{A}} \mathbb{E}_{s_0 \sim \rho_0, s_{t+1} \sim T(s_t, a_t, \cdot)} \left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t, s_{t+1}) \right] \quad (1)$$

which maximizes the expected cumulative reward by selecting optimal actions $a_0 \dots a_{H-1} \in \mathcal{A}$.

While the vehicular control decision processes that we consider in this paper may instead be considered as infinite-horizon MDPs, which maximizes the expected cumulative reward for $H \to \infty$, practical methods often optimize over a large finite H as a proxy for $H \to \infty$.

B. Policy-Based Model-Free Deep Reinforcement Learning

Policy-based model-free DRL algorithms define a policy $\pi_\theta(a|s)$ which gives the probability of taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$. The policy is parameterized by θ (*e.g.* weights in a linear function or neural network), which is optimized so that the policy maximizes the expected cumulative reward

$$\max_\theta \mathbb{E}_{\substack{s_0 \sim \rho_0, a_t \sim \pi_\theta(\cdot|s_t) \\ s_{t+1} \sim T(s_t, a_t, \cdot)}} \left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t, s_{t+1}) \right] \tag{2}$$

The REINFORCE policy gradient algorithm [44] maximizes the expected cumulative reward of policy π_θ in Equation 2 by sampling trajectory $(s_0, a_0 \dots, s_{H-1}, a_{H-1}, s_H)$ and optimizing the following objective

$$\arg\max_\theta \sum_{t=0}^{H-1} \sum_{t'=t}^{H-1} \gamma^{t'-t} r(s_{t'}, a_{t'}, s_{t'+1}) \tag{3}$$

via gradient descent on θ. At each update step

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(\cdot|s_t)|_{a_t} \sum_{t'=t}^{H-1} \gamma^{t'-t} r(s_{t'}, a_{t'}, s_{t'+1}) \tag{4}$$

where α is the learning rate.

Like REINFORCE, the Trust Region Policy Optimization (TRPO) algorithm [42] also collects a trajectory and optimizes the objective in Equation 3. However, to encourage training stability, TRPO constrains the update step of θ so that the policy does not change too quickly:

$$\begin{aligned} \theta \leftarrow \arg\max_{\theta'} & \sum_{t=0}^{H-1} \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \sum_{t'=t}^{H-1} \gamma^{t'-t} r(s_{t'}, a_{t'}, s_{t'+1}) \\ \text{subject to } & \frac{1}{H} \sum_{t=0}^{H-1} D_{\mathrm{KL}}(\pi_\theta(\cdot|s) \| \pi_{\theta'}(\cdot|s)) \le \delta_{\mathrm{kl}} \end{aligned} \tag{5}$$

where δ_{kl} is the upper bound of the mean KL divergence between the updated policy $\pi_{\theta'}$ and the original policy π_θ, preventing θ' from deviating too far from θ. In practice, TRPO eliminates the need to tune the learning rate α, which is a sensitive hyperparameter while δ_{kl} is a standard constant.

We note that unlike our previous works [4], [16], we solely learn a policy and do not additionally learn a value function. Many actor-critic and value-based algorithms learn a value function or Q-value function to fit the cumulative reward at a given state or a given state-action, respectively. While these algorithms demonstrate improved performance in certain applications, properly learning a value function requires selection of the optimizer (*e.g.* ADAM [45], RMSprop [46]), tuning of the learning rate, tuning of the smoothing hyperparameter λ for generalized advantage estimation [47], tuning of the hyperparameter for value clipping, and other potentially difficult algorithmic choices. In multi-vehicle domains in particular, fitting the value function to the system objective likely requires the value function to operate over the entire state at a given time, which is difficult and inflexible to encode, while we specify in Sections V-B and V-C that our policy-based methodology only requires local observability.

IV. Automated Vehicular Systems

We describe the general types of simulated vehicular systems compatible with our methodology for automatic vehicle control. Overall, we focus on microscopic simulations which consider the interactions of individual vehicles rather than aggregate behavior of traffic flow. We require the ability to repeatedly run simulations for a duration from a set of initial simulation states. Each simulation evolves the positions and velocities of the vehicles through time following defined physical rules. We assume that every vehicle in the system follows its own route, which is assigned by some fixed algorithm given the origin and destination; we do not consider decision-making for route assignment in this work. In *closed* systems, vehicles circulate within the system endlessly, following assigned routes. In *open* systems, vehicles enter the systems (*inflow*) at their origins and exit the systems (*outflow*) at their destinations. Within each system, a fraction or all of the vehicles are automated and can be controlled in some manner, while the rest of the vehicles follow modeled default behavior; each system must have one or more automated vehicles. We assume that a central objective exists and can be quantified for the system; for example, the objective could be a function of vehicle speeds, system throughput, fuel consumption, or safety in the system. Note that even for system objectives purely based on speeds or throughput, attempting to control each *individual* vehicle towards the maximum speed possible could often be suboptimal due to negative, congestion-inducing effects on surrounding vehicles.

We briefly describe two such automated vehicular systems:

1) Traffic Systems: Mixed autonomy traffic systems where fractions of all vehicles are automated are studied in [4], [5], [13]–[17], [41]. The acceleration and lane change (if applicable) decisions of AVs may be controlled. Uncontrolled vehicles follow default behavior dictated by well-studied car-following models, such as the Intelligent Driver Model (IDM) [48].

2) Robotic Warehouse Systems: A warehouse management system for hundreds of cooperatively controlled mobile robots with origins and destinations is studied in [3]. In this system, routing and movement of robots comprise a challenging joint control task which is often decoupled into dynamic route assignment followed by movement planning. Dynamic route assignment is addressed by [8], [49] and other methods, while our methodology for automated vehicle control applies to the movement planning problem. Acceleration of the robots may be controlled. Strategies resembling mixed autonomy may reduce the computational complexity: control may be restricted to the set of robots making critical decisions at a given time while non-critical robots can follow default behaviors (*e.g.* move at the maximum speed possible).

Other examples of such systems include metro train regulation [9] and airport surface management [10].

In this article, we validate the methodology on traffic systems. In practice, each system may support a variety of vehicle densities. Therefore, we desire policies which generalize across multiple *configurations* of vehicle density.

V. Deep Reinforcement Learning Methodology

We prescribe a unified DRL methodology for automatic control of vehicular systems. An important contribution of this work is to minimize the DRL-related design choices that a researcher or practitioner has to make. Here we describe the common components of the DRL methodology necessary for all vehicular systems.

A. MDP Definition

We naturally model a vehicular system in microscopic simulation as a MDP. At each time step t, the state s_t is composed of the positions, velocities, and other metadata of all vehicles in the road network. The action a_t is the tuple of per-AV actions of all AVs in the road network at step t. The reward function $r(s_t, a_t, s_{t+1})$ is specified so that the cumulative reward is the objective. A key distinction from our previous works is that the reward function does not need to be manually shaped by the researcher or practitioner to encourage behaviors attaining higher objective values. The stochastic transition function T is not explicitly defined, but $s_{t+1} \sim T(s_t, a_t, \cdot)$ can be sampled from the microscopic simulator, which applies the actions for all vehicles over a simulation step duration δ. In simulations which do not protect against vehicle collisions, we introduce a collision penalty $-\lambda_{\text{collision}} n_{\text{collision}}$ to the reward function $r(s_t, a_t, s_{t+1})$ where $n_{\text{collision}}$ is the number of collided vehicles in s_{t+1} and $\lambda_{\text{collision}}$ is large enough to discourage the AVs from collision-inducing behaviors.

B. Partial Observability

In practice, as the state s could be large or difficult to reason about, DRL methods often approximate the policy with $\pi_\theta(\cdot|s) \approx \pi_\theta(\cdot|o)$ [42], where the *observation* $o = z(s) \in \mathcal{O}$ possesses only a subset of the information of the state s, z is the observation function, and $\mathcal{O}$ is the observation space. Together, $(\mathcal{M}, \mathcal{O}, z)$ actually defines a partially observable MDP (POMDP) [50]. Partial observability is natural in real-world decision processes since obtaining a local observation may be more feasible than the full state. For example, an AV may more easily observe nearby vehicles than faraway vehicles, and information regarding faraway vehicles may not help the decision algorithm much anyways. Guided by these principles, we design the observation function for systems with a single AV to include only the entities most relevant to the AV's decision; systems with multiple AVs will be considered next. The observation function is one of the few design choices made by the researcher or practitioner for the methodology presented within this work.

C. Multi-Agent Policy Decomposition

In vehicular systems with multiple AVs, we apply multi-agent policy decomposition with each AV as an *agent*. A MDP with multiple action dimensions could naturally be formulated as a decentralized partially observable MDP (Dec-POMDP) [51]. In this case, we refer to the action space of the original MDP as the joint action space, which factorizes into the product of M agent action spaces in the Dec-POMDP framework. The policy $\pi_\theta(a|s)$ decomposes into per-agent policies $\pi_\theta(a^i|o^i)$ such that $\pi_\theta(a|s) = \prod_{i=1}^{M} \pi_\theta(a^i|o^i)$, where $a^i \in \mathcal{A}^i$, the action space of agent $i \in \{1, \cdots, M\}$, and $o^i = z(s, i) \in \mathcal{O}^i$, the observation space for agent i. We have $o^1 \cup \cdots \cup o^M \subseteq s$ and $\mathcal{A}^1 \times \cdots \times \mathcal{A}^M = \mathcal{A}$. z is a defined observation function which maps state s to observation o^i for agent i. Without multi-agent policy decomposition, the combinatorial nature of $\mathcal{A}$ poses an intractable problem to learning algorithms. Like in single-AV systems, z must be designed by the researcher or practitioner.

D. Per-AV Action Space

The longitudinal per-AV action space can often be naturally formulated as a continuous acceleration space $\mathcal{A}^i_{\text{longitudinal}} = [-c_{\text{decel}}, c_{\text{accel}}]$ for each AV i. However, in systems where multiple AVs interact, we prescribe a discrete bang-off-bang acceleration space $\mathcal{A}^i_{\text{longitudinal}} = \{-c_{\text{decel}}, 0, c_{\text{accel}}\}$, which we find to empirically improve coordination between multiple AVs despite forgoing fine-grained control. For systems which require AVs to make lateral (lane change) decisions, the lateral action space is the set of lane indices $\mathcal{A}^i_{\text{lateral}} = \{1, \ldots, L\}$ to travel in, where L is the number of lanes. Therefore $\mathcal{A}^i = \mathcal{A}^i_{\text{longitudinal}} \times \mathcal{A}^i_{\text{lateral}}$ for systems permitting AV lane change and $\mathcal{A}^i = \mathcal{A}^i_{\text{longitudinal}}$ otherwise.

E. Per-AV Policy Architecture

We define the per-AV policy $\pi_\theta(a^i|o^i)$ as a neural network with three fully-connected layers with a hidden size of 64. If there are multiple AVs, we share the policy parameter θ across all vehicles in the traffic network to share experiences between AVs [52]. For systems requiring joint action for each AV (*i.e.* $\mathcal{A}^i = \mathcal{A}^i_{\text{longitudinal}} \times \mathcal{A}^i_{\text{lateral}}$), the policy is a neural network with multiple heads, one for each factor of the joint action.

F. Reward Centering and Normalization

To reduce variance of policy gradients [53], we apply reward centering and normalization to the original reward before calculating the objective for policy gradient

$$r_{\text{norm}}(s_t, a_t, s_{t+1}) = \frac{r(s_t, a_t, s_{t+1}) - \hat{\mu}_r}{\hat{\sigma}_R} \tag{6}$$

where $\hat{\mu}_r$ is the running mean of r and $\hat{\sigma}_{\hat{R}}$ is the running standard deviation of the running cumulative reward, which is updated according to $\hat{R} \leftarrow \gamma \hat{R} + r(s_t, a_t, s_{t+1})$.

G. Multi-Task Learning Over Configurations

As we consider multiple configurations with varying vehicle densities for each vehicular system, training a separate policy for each configuration would be cumbersome and inefficient. Thus, we discretize the density configuration space

into equally-spaced density configurations and learn a single multi-task policy over this configuration set. During training, we initialize separate environments for each configuration in the configuration set. At each training step, our policy gradient algorithm receives trajectories from all environments and batches the gradient update due to these trajectories. Multi-task learning allows a single trained policy to generalize across a range of configurations, avoiding the costs of training a separate policy for each configuration.

H. Derived Policies

We extract the behaviors discovered by DRL policies by hand-designing simple rule-based policies with one or two optimized parameters. We denote these policies as the *Derived* policies because they are grounded in the DRL policies' behaviors. The purpose of constructing Derived policies is two-fold: 1) the Derived policies offers a comparison between the DRL policy and a gold-standard policy which shares the similar behavior 2) the Derived policies are easily interpretable and may be analyzed further for practical deployment. The steps we take to construct a Derived policy are as follows:

1) Given a traffic system, train a performant DRL policy over a desired range of density configurations.
2) For each density configuration, examine the behavior of the DRL policy through videos and time-space diagrams, noting common behaviors shared across multiple density configurations.
3) Formulate these behaviors into an interpretable rule-based parameterized policy across density configurations.
4) For each density configuration independently, obtain the optimal parameters of derived policy with exhaustive grid search, which is feasible and straightforward for low-dimensional parameter spaces.

To ease the hand-design process, we permit Derived policies to use information from any part of the state, contrasting with DRL policies which must arrive at decisions based on observed information only and must generalize well across all density configurations. When possible, we identify optimal parameter values that may be shared across ranges of densities configurations.

I. Complexity Analysis

We analyze the computational complexity of our methodology for vehicular systems with M AVs. Our methodology decomposes the action space $\mathcal{A}$ into the product of agent action spaces $\mathcal{A}^i$ and restrict state space $\mathcal{S}$ into corresponding agent observation spaces $\mathcal{O}^i$. For our analysis, we make the assumption that the design of observation function z preserves sufficient observability for decision making and that the decisions of other AVs do not induce non-stationarity. The former assumption may be realistic if observation functions are designed to exclude distant and non-causal facets of the state which may have little impact on the current decision. The latter assumption may be realistic if the policy changes slowly across updates, if individual acceleration actions have small effects on the state, and if the AV penetration rate is low. These assumptions factors the joint control problem to M independent control problems, each with state space $\mathcal{O}^i$ and action space $\mathcal{A}^i$.

The TRPO algorithm leveraged by our methodology is a variation of the Natural Policy Gradient (NPG) algorithm [54], [55], so we proceed to invoke the computational complexity of NPG at training time. As proved by [55], NPG obtains an ϵ-optimal policy in tabular RL settings with no more than $\frac{2}{(1-\gamma)^2\epsilon}$ gradient update steps, and each update step in the tabular case takes $O(|\mathcal{O}^i||\mathcal{A}^i|)$ time. In our work, we leverage functional approximation with a neural network, enabling tractable parameter updates though losing optimality guarantees of the tabular case. Updating the parameters of the neural network with a forward and backward pass of the neural network takes $f(|o|, |a|)$ time in general, where $o \in \mathcal{O}^i$ and $a \in \mathcal{A}^i$ are individual observation and action vectors, respectively. In practice, rather than defining an ϵ and taking $\frac{2}{(1-\gamma)^2\epsilon}$ gradient update steps, we take a total of G gradient update steps sufficient for the training performance to converge while batching over B sampled trajectories for each gradient update, horizon H simulation steps per trajectory, and M agents at every simulation step, for a total training time complexity of $O(GBHMf(|o|, |a|))$. When executing an already trained model to generate a trajectories for H simulation steps, the execution time complexity is $O(HMf(|o|, |a|))$.

The $f(|o|, |a|) = O(|o| + |a|)$ computations introduced by small fully-connected networks such as ours may be small in practice compared to other factors such as inter-process communication overheads and the simulation time of the system itself, which is $O(GBHN)$ at training time and $O(HN)$ at execution time, where $N \geq M$ is the total number of vehicles in the system. However, if more computationally intensive convolutional, recurrent, or attention-based neural networks are necessary, the neural network computation time may increase significantly, requiring either more parallelism in the forms of GPUs or more efficient feature engineering and architecture design to reduce computational cost.

Mixed autonomy ($M < N$) directly reduces computational complexity compared to full autonomy ($M = N$) by reducing M, while also indirectly reducing the complexity by lowering non-stationarity present in the system, permitting the usage of smaller G, B, and H factors; in other words, in full autonomy systems the outcome of an AV's decision is more likely to be affected by decisions made by other AVs, where all nearby vehicles are AVs. Nevertheless, mixed autonomy control may sacrifice some degree of optimality compared to full autonomy, due to underactuation.

VI. Experimental Setup

We describe the general simulation, training, and evaluation setups of our experiments. We provide reference ranges here and reserve full details for the code.

A. Vehicular Systems

We construct six diverse mixed autonomy traffic systems in the SUMO microscopic simulator [56] to demonstrate the generality of our unified methodology. Three systems are open

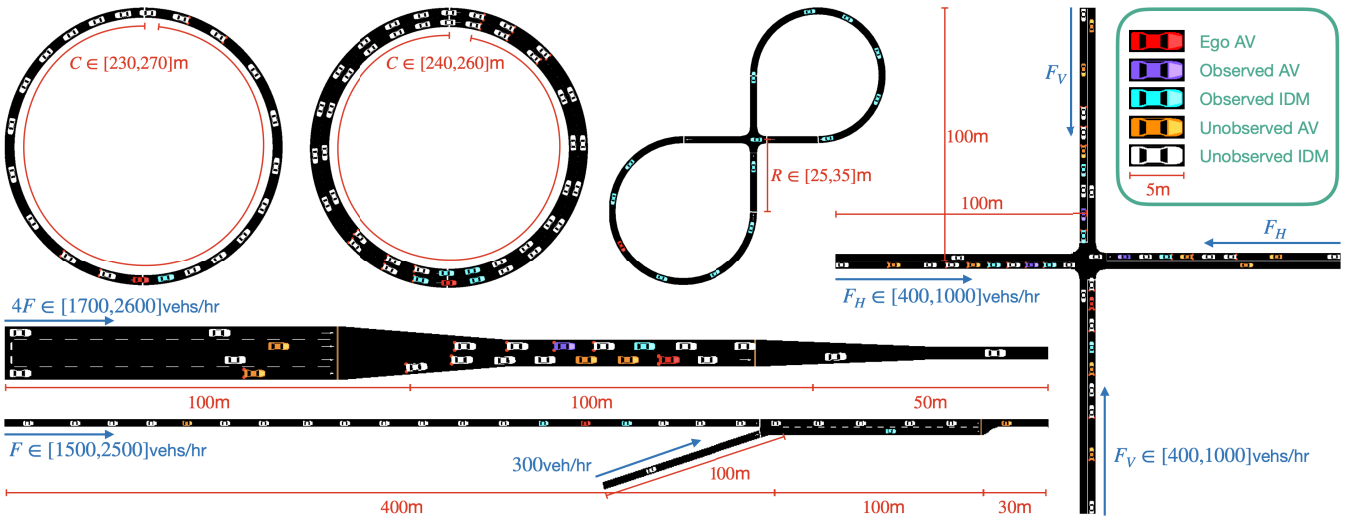


Fig. 2. **Experimental traffic systems.** In clockwise order from the top left: Single Ring, Double Ring, Figure Eight, Intersection, Highway Ramp, and Highway Bottleneck. Each traffic system is independently drawn to scale. Single Ring, Double Ring, and Figure Eight are closed systems with 22, 44, and 14 vehicles respectively. Intersection, Highway Ramp, and Highway Bottleneck are open systems with variable numbers of vehicles. Within each system, we designate one AV as the ego AV (red) without loss of generality and color other vehicles according to their type and whether they are observed by the ego AV. Each AV typically observes the speed, relative position, and type (AV or uncontrolled) of itself and its observed vehicles.

and three systems are closed. While we do not incorporate any traffic control element, such as traffic light or ramp meter, future work may easily incorporate these traffic control elements as needed in conjunction with mixed autonomy control.

Common to all systems, all vehicles are 5m in length and uncontrolled vehicles follow the IDM with a Gaussian acceleration noise of 0.2m/s^2. A randomized initialization is obtained by simulating H_0 warmup steps starting from an arbitrary set of vehicle positions; the next H steps are measured for performance. We use full SUMO safety checks, which prevent vehicles from entering most collisions situations. For each system, we consider multiple configurations of traffic densities.

In closed systems, we define the objective to be the total cumulative distance traveled by all vehicles, which is proportional to the average speed over all vehicles over all timesteps. We use a simulation step size of $\delta = 0.1$s for all closed systems and terminate the simulation immediately if an occasional collision occurs despite the safety check. The density configuration is varied by scaling the traffic network geometries while holding the number of vehicles constant.

In open systems, we designate the objective to be the throughput (outflow per hour) of the system. We use a simulation step size of $\delta = 0.5$s and do not terminate the simulation if two vehicles collide: only the collided vehicles are removed from the simulation and do not count towards the outflow. Each density configuration corresponds to a target inflow rate (vehicles per hour), which controls the number of vehicles in the system. If a vehicle is not able to inflow due to congestion in an inflow lane, the vehicle is dropped from simulation. Unlike in closed systems, the number of vehicles in open systems is not constant and depends on the inflow rate.

As we already examine each traffic system under multiple traffic density configurations, we do not perform additional ablation on the effect of the AV penetration rate and instead choose to fix a penetration rate for each system. We refer readers to several previous works for ablations on the effect of penetration rate [14], [16]. While all of our systems contain under 50 vehicles at any given time, our methodology naturally extends to much larger systems if two conditions hold: 1) the objective (*e.g.* throughput) can be decomposed into local objectives (*e.g.* local throughput), and 2) the total number of decisions, which can be reduced via pruning, is not exorbitantly high.

We name and describe each traffic system, along with our constructed observation function. To encourage AVs to develop generalizable behaviors based on local information, we do not allow AVs to observe the underlying traffic density configuration parameter. All traffic systems and corresponding observation spaces are visualized in Figure 2.

1) Single Ring (Closed): The Single Ring system consists of 22 vehicles in a single-lane ring network with circumference $C \in [230, 270]$ m; each C corresponds to a density configuration. We designate one vehicle as AV while leaving the 21 other vehicles uncontrolled. The AV's observation function z consists of the AV's speed and the offset and speed of the leading vehicle. We consider two differing objectives:

a) Global: The objective is the cumulative distance traveled by all vehicles in the simulation. The reward function $r(s, a, s')$ is therefore the average speed of all vehicles in s'.

b) Greedy: The objective is the cumulative distance traveled by the AV. The reward function $r(s, a, s')$ is therefore the AV's speed in s'.

2) Double Ring (Closed): The Double Ring system consists of 44 vehicles in a two-lanes ring network with circumference $C \in [240, 260]$ m. The SUMO simulator does not account for the exact geometry of the road and instead simulates the inner lane and outer lane to be the same length. We designate one vehicle in the outer lane as the AV, leaving the

43 other vehicles uncontrolled. In addition to controlling its own acceleration, the AV is allowed to change lane; no other vehicle is allowed to change lane. The observation function z includes the speed and lane index of the AV and the speeds and offsets of the leading and following vehicles in both lanes. Like the Single Ring, we consider two cases corresponding to the {Global, Greedy} reward functions.

3) Figure Eight (Closed): The Figure Eight system consists of 14 vehicles in a closed single-lane two-way intersection network. Each direction (westbound or northbound) of the two-way intersection consists of length $R \in [25, 35]$ m straightaways before and after the intersection; each R corresponds to a density configuration. The two directions are connected by 270° circular arcs. We designate one vehicle as AV, leaving others uncontrolled. The AV's observation function z consists of the distance from the intersection and speed for every vehicle, reflecting the symmetry of the two loops. The reward function $r(s, a, s')$ is the average speed of all vehicles in s'.

4) Highway Bottleneck (Open): The Highway Bottleneck system simulates a straight highway with four 100m-long inflow lanes which merge into two 100m-long lanes then merge into a single 50m-long lane, from which vehicles outflow. All four inflow lanes share a per-lane target inflow rate of F; so the total target inflow rate is $4F \in [1700, 2600]$ vehs/hr. At the first merge (four lanes to two lanes), the top two lanes merge together and the bottom two lanes merge together. No vehicle may change lane. We designate 20% of the vehicles as AVs. Let the *merge lane* be the lane which merges with the AV's lane. The observation function for each AV is the speed and the distance to the next merge of the AV, the offset and speed of closest following AV on the merge lane, and the offset and speed of closest following uncontrolled vehicle on the merge lane.

5) Highway Ramp (Open): The Highway Ramp system simulates a straight single-lane highway with an on-ramp. The single-lane highway proceeds for 400m when it meets a 100m on-ramp to form 100m of a two-lane merging region. The two-lanes merge into a single lane at the end of the 100m merging region, and the single-lane highway continues for another 30m. The highway sees a target inflow rate $F \in [1500, 2500]$ vehs/hr while the ramp sees a target inflow rate of 300 vehs/hr. No vehicle may change lane. We designate 10% of the highway vehicles as AVs, leaving the rest uncontrolled, including all ramp vehicles. The observation function for each AV is the speed of the AV, the offsets and speeds of the leading and following vehicles on the highway, and the offset and speed of the following vehicle on the ramp.

6) Intersection (Open): The Intersection system simulates a single-lane intersection with inflows and outflows in each cardinal direction. The intersection only permits straight traffic and does not permit turns. Along each direction, the intersection is situated between two 100m long road segments. We consider configurations of pairs of horizontal and vertical target inflow rates $F_H, F_V \in [400, 1000]$ vehs/hr; configurations with $F_H + F_V < 1400$ vehs/hr are excluded due to trivially low inflow. We designate 33% of the vehicles as AV. The observation function for each AV includes the position and speed of the heads and tails of the closest *chains* to the intersection, where we define each *chain* to be an AV and any uncontrolled vehicles that it immediately leads. The rationale behind this design is that each AV may provide control to all tailing uncontrolled vehicles.

B. Baseline Policies

For each system, we define the Baseline policy to follow the SUMO IDM behavior for all AVs. As collisions may frequent occur in the Figure Eight and Intersection systems under the Baseline policy, the vertical directions are given priority over the horizontal directions, which must slow to a near-stop before proceeding.

For the Single Ring, Highway Bottleneck, and Intersection systems, we adjust DRL algorithms from prior works [4], [5], [17] to train policies within our respective traffic systems, which are similar but may differ somewhat in construction from the those from the prior works. For these algorithms, we use the exact same training and evaluation setup as described below for our own methodology when applicable to ensure fairness of comparison.

C. Training

For each system, we train a policy for up to $G = 200$ gradient update steps with the TRPO algorithm. We perform each gradient step with the batched data from $40 \leq B \leq 45$ collected trajectories, divided among equally-spaced configurations. For each trajectory, we use $H_0 \leq \frac{100}{\delta}$ warmup steps and horizon $H = \frac{1000}{\delta}$; warmup steps provide randomness in the MDP initialization. Unlike typical model-free DRL setups which may sweep over many DRL algorithms each with many hyperparameters involved in training the policy or value function, the only tuned hyperparameter in this article is the discount factor $\gamma \in [0.9, 0.9999]$, where $1 - \gamma$ is searched in log-space. Training each policy takes less than 3 hours on an Intel Xeon Platinum CPU machine with 48 cores. Though training is stochastic, we do not observe significant variations in learned behavior and performance between runs. For systems much larger than ones considered in this work, TRPO may result in slow training and high memory consumption and may be replaced with REINFORCE.

D. Evaluation

For each system, we select the checkpoint with the best average objective value on the batched training trajectories to evaluate. To evaluate the checkpoint on each configuration of each system, we sample 10 trajectories with different initial seeds. To allow traffic dynamics to achieve steady state, we use longer $H_0 \leq \frac{500}{\delta}$ warmup steps, sufficient to allow congestion to fully build up under the Baseline policy. We then run the policy for $H_1 \leq \frac{1500}{\delta}$ steps to allow traffic dynamics to achieve steady state under the evaluated policy, before measuring the objective value (speed or outflow) on a last $H \leq \frac{1000}{\delta}$ steps. The choice of H_0, H_1, and H are not significant as long as H_0 and H_1 are each long enough for traffic dynamics to achieve steady state.

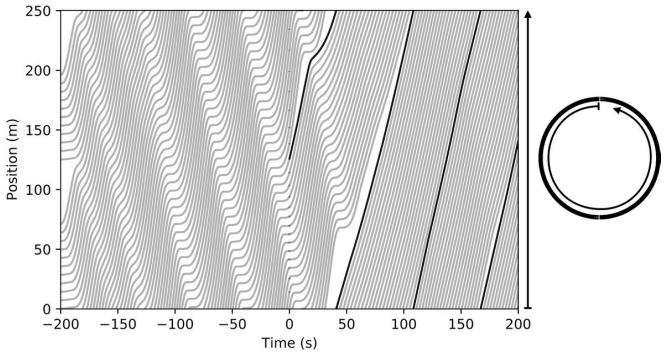


Fig. 3. **Single Ring** $C = 250$ m **time-space diagram.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned Global policy (on and after time 0s). Bold indicates the AV controlled by the DRL policy. Arrows indicate progression of vehicles. The DRL policy controls the AV to eliminate the backward propagating waves formed under the Baseline policy.

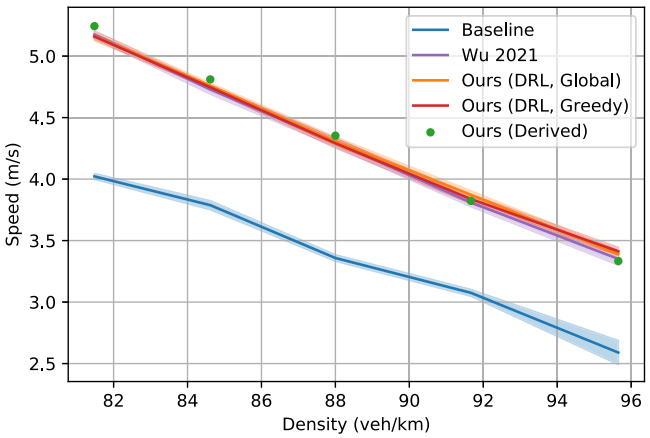


Fig. 4. **Single Ring average speed.** We compare the average speed over all 22 vehicles over horizon H under the Baseline, DRL, and Derived policies, with the shaped deep reinforcement learning policy Wu 2021 [4] as additional comparison. Our DRL and Derived policies see significantly better average speeds than those of the Baseline policy. We display the Derived performance as points instead of a single line because the optimal speed parameter v_{target} are not shared for any of the density configurations. We show that our unified methodology produces similar performances to Wu 2021 without hand-designed acceleration penalties which encourage convergence to a constant speed.

VII. Experimental Results

We present both numerical performance comparisons with the Baseline policy as well as a behavioral dissection, visualized via time-space diagrams, for representative configurations of each traffic system studied. We measure numerical performances after sufficient duration has passed for vehicle dynamics to achieve steady state. In all systems, we demonstrate that DRL discovers interesting and sometimes surprising behaviors which significantly outperform the Baseline. To shed light on the performant behaviors discovered automatically via DRL, we extract DRL behavior into rule-based Derived policies and offer numerical performance comparisons. For all speed and outflow results, we compute the means and standard deviations across 10 trajectories with different seeds; the standard deviation may sometimes be small for the Baseline and Derived policies. We reserve experiments demonstrating robustness of Derived policies to different car following model parameters for Appendix A.

A. Single Ring

Due to the linear string instability of IDM [57], the Baseline policy quickly results in a stop-and-go waves which propagate in the opposite direction of traffic [58] under all density configurations. Under both the Greedy and Global policies, the AV learns to mitigate stop-and-go waves in every configuration by converging to a constant speed. We illustrate the Baseline and Global behaviors in Figure 3.

Mimicking this behavior, we design a Derived policy with a single, optimized target speed v_{target} per circumference configuration. Figure 4 compares the average speeds among the DRL, Derived, and Baseline policies. The DRL policies nearly matches the Derived policies despite seeing local observations only, without knowledge of the true circumference configuration. Our results here are similar to Wu 2021 [4] (Figure 4) with one important difference: the prior work utilizes an additional acceleration penalty to encourage convergent behavior in speed while we show that a simple speed-based objective alone is sufficient for DRL to discover convergent behavior. In addition, [4] only considers a global objective, while we consider both global and greedy objectives.

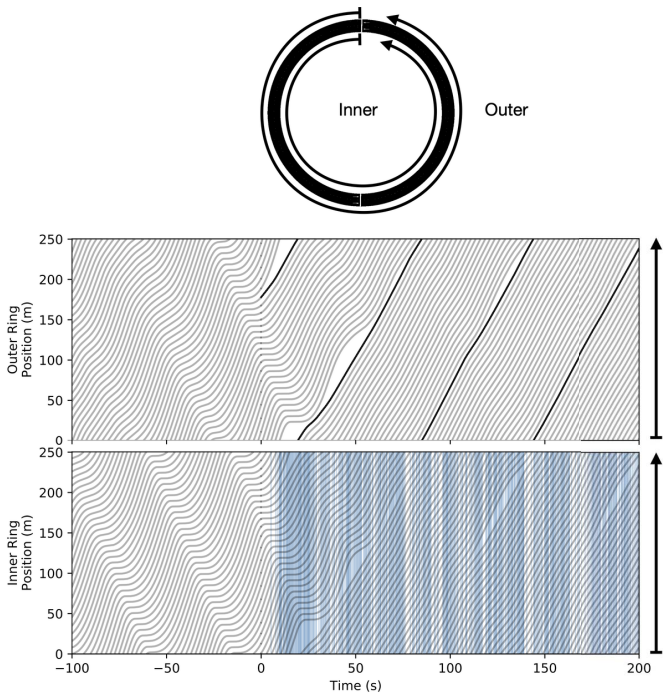
Algorithm 1 Single Ring Derived Policy

procedure DERIVED(s) ▷ State s
  $C \leftarrow$ get circumference from s
  $v_{\text{target}} \leftarrow$ tuned target speed parameter for C
  $v \leftarrow$ get speed of the AV from s
  return EQUALIZE(v_{target}, v)

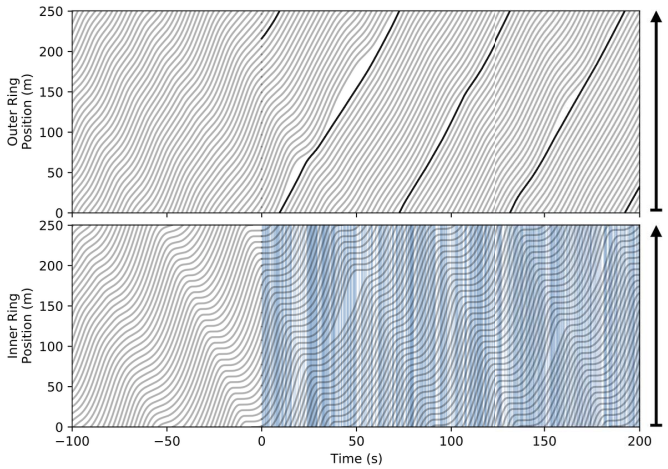
procedure EQUALIZE($v_{\text{target}}, v_{\text{current}}$)
  if $v_{\text{current}} < v_{\text{target}}$ **then return** $0.75\ c_{\text{accel}}$
  else if $v_{\text{current}} > v_{\text{target}}$ **then return** $-0.75\ c_{\text{decel}}$
  else return 0

B. Double Ring

Under the Baseline policy, each lane in the Double Ring exhibits identical behavior to the Single Ring. However, equipping the AV with the ability to change lane results in differing behaviors when maximizing the Greedy or Global objective with DRL. The Greedy policy learns to stay and converge to a constant speed within its own (outer) lane while disregarding the vehicle movement in the inner lane completely. On the other hand, the Global policy learns to mitigate the stop-and-go waves within both lanes *simultaneously* by converging to a constant speed within its own lane while *flashing the turn signal to regulate the speed of the inner lane without physically changing lane*. The behaviors are shown in Figure 5 and compared numerically in Figure 6. We note that the AV under the Greedy policy also frequently flickers its turn signal, as seen in Figure 5; further investigation is required to differentiate the signal patterns of the two policies, which leads to significant differences in performance outcomes. Though this particular Global behavior exploits a flaw in the SUMO simulation, we note that a naturalistic human driver may also



(a) Global Policy



(b) Greedy Policy

Fig. 5. **Double Ring** $C = 250$ m **time-space diagrams.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned Global or Greedy policies (on and after time 0s). Bold indicates the AV controlled by the DRL policy. Arrows indicate progressions of vehicles in the outer and inner lanes. In both Global and Greedy, the DRL-controlled AV eliminates the backward propagating waves that form under the Baseline policy within its own lane. Turn signal flickering (blue vertical strips) by the Global policy strategically mitigates the waves that form in the *other* lane, while that of the Greedy policy does not.

slow down if a leading vehicle in another lane attempts to change lane into the space ahead. We construct the Derived policy in an identical manner to the Single Ring, without lane change. However, our Derived policy lacks the strategic turn signal behavior of Global.

C. Figure Eight

As the intersection is unsignalized, the Figure Eight system under the Baseline policy sees vehicles alternating to pass the

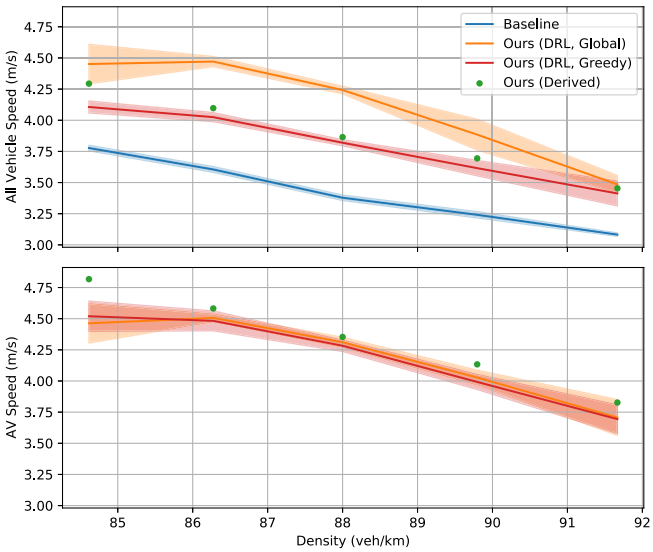


Fig. 6. **Double Ring average and AV speed.** Over horizon H, we compare the average speed over all 44 vehicles (top) and AV speed (bottom) under the Baseline, DRL, and Derived policies. The Global policy sees the best average speed in almost all cases, due to mitigation of stop-and-go waves in both lanes. The Derived and Greedy policies may see better AV speed than Global due to better mitigation of waves within the AV's own lane.

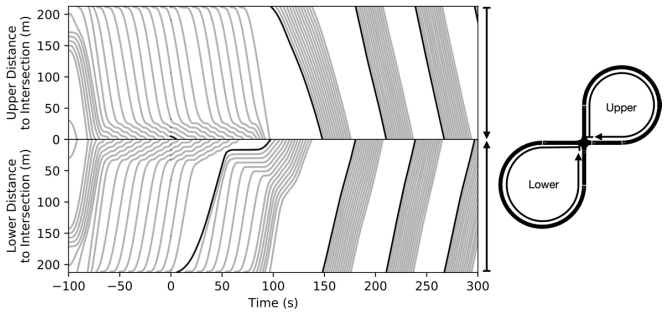


Fig. 7. **Figure Eight** $R = 30$ m **time-space diagram.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Bold indicates the AV controlled by the DRL policy. Arrows indicate progressions of vehicles approaching the intersection from the upper and lower loop. The AV guides a snaking behavior that eliminates alternation of single vehicles at the intersection.

intersection one by one, similar to the behavior at a stop-sign. As shown in Figure 7, the DRL policy instead learns to slow down to gather the rest of the vehicles as followers, then increases the speed while the other vehicles follow to "snake" around the Figure Eight. This behavior allows the speed of all vehicles to be faster than the average Baseline speed, as shown in Figure 8. Using the same approach as the Single Ring, we design the Derived policy by applying exhaustive search to find an optimal target speed v_{target}. We find that DRL achieves close to the tuned target speed for all configurations.

While [15] reports similar DRL behavior in the Figure Eight systems, it shapes the reward function to explicitly encourages convergence towards a handpicked target speed. On the other hand, our present work demonstrates that simply optimizing for the end objective suffices without any handcrafting by the researcher or practitioner.

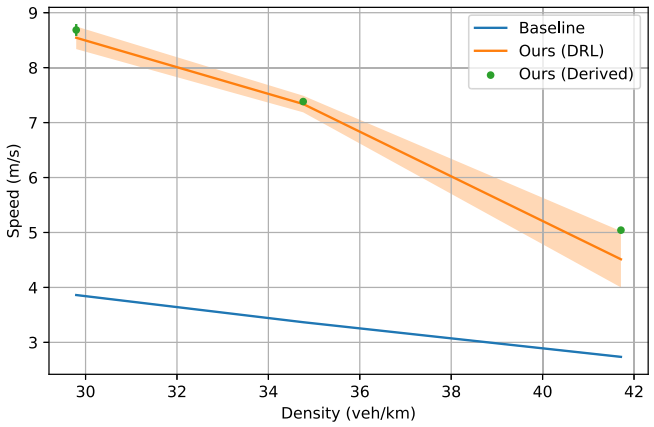


Fig. 8. **Figure eight average speed.** We compare the average speed over all 14 vehicles over horizon H under the Baseline, DRL, and Derived policies. The DRL policy nearly matches the Derived policy, despite needing to infer the target speed from solely the observations.

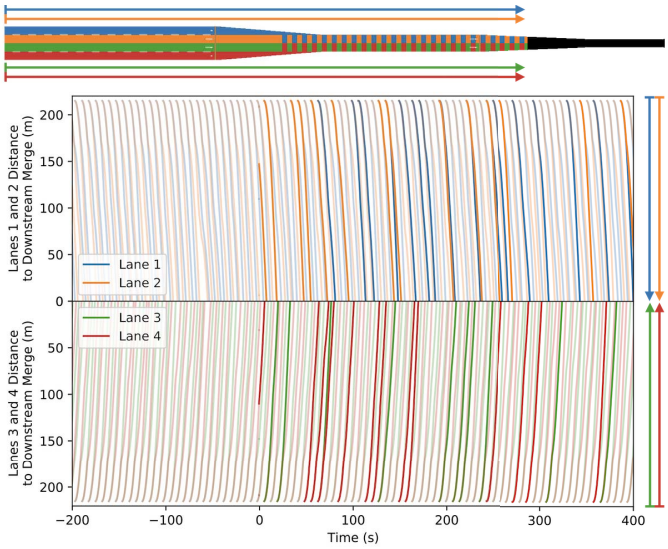
Algorithm 2 Figure Eight Derived Policy

procedure DERIVED(s) ▷ State s
  $R \leftarrow$ get radius from s
  $x \leftarrow$ total distance of the figure eight
  $x_{\text{last}} \leftarrow$ distance from the last follower to the AV
  if $x_{\text{last}} < \frac{x}{2}$ **then**
    $v_{\text{target}} \leftarrow$ tuned target speed for R
  else ▷ Slow initial speed to gather followers
    $v_{\text{target}} \leftarrow 0.5$m/s
  $v \leftarrow$ get speed of the AV from s
  return Equalize(v_{target}, v)

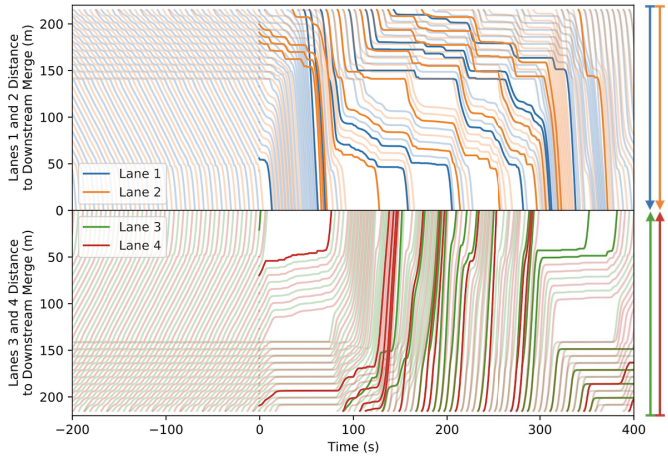
D. Highway Bottleneck

The Highway Bottleneck under the Baseline policy sees two distinct behaviors: at low target inflow rates $F < 2200$ vehs/hr, vehicles from the two merging lanes may weave together without slowing down; at high target inflow rates $F \geq 2200$ vehs/hr, a capacity drop phenomenon [59] occurs, and vehicles from the two merging lanes slow down to a near stop before the merge, taking turns to continue onto the merged lane. We observe that the behavior (Figure 9) of the trained DRL policy is similar to Baseline for $F < 2200$ vehs/hr; however, AVs learn to reduce alternation at merge points for $F \geq 2200$ vehs/hr, achieving higher throughput by letting a group of vehicles pass at once (Figure 10).

We consider the DRL method introduced by Vinitsky 2018 [5] as an additional baseline in Figure 10. While this prior work reduces the control space of the policy to upstream segments of the highway bottlenecks to encourage ramp metering behaviors, our methodology does not impose artificial restrictions to guide the policy. To train Vinitsky 2018, we augment the original method described by [5] with the RMSprop optimizer [46], which we find to improve performance over the ADAM optimizer [45] used by [5]. Our DRL policy performs similarly on average to Vinitsky 2018, with better performance for lower F and worse performance for higher F. These trade-offs in performance



(a) $F = 2000$ vehs/hr



(b) $F = 2400$ vehs/hr

Fig. 9. **Highway Bottleneck time-space diagrams.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Blue, orange, green, and red lines indicate vehicles originating on lanes 1, 2, 3, and 4, respectively, and correspond to colored arrows indicating progressions of vehicles. Bold indicates the AVs controlled by the DRL policy. For $F < 2200$ vehs/hr, DRL sees the same efficient behavior as the Baseline. For $F \geq 2200$ vehs/hr, Baseline degrades significantly into an inefficient alternation, DRL reduces alternation by letting groups of vehicles pass the downstream bottleneck at once.

suggests that an interesting topic of future research may study the advantages and limitations of an unified methodology for segment-based control of mixed autonomy traffic.

For additional comparison, we design a Derived policy with tuned threshold parameters x_1 and x_2 which attempts to reduce alternation in a similar way to our policy if $F > 2200$ vehs/hr, otherwise mimicking Baseline behavior. Essentially, AV i stops near the merge point if the following vehicle on the adjacent lane is uncontrolled and also near the merge point. This encourages AV i to wait until the vehicle on the adjacent lane is an AV before continuing. The Derived policy suffers more at $F = 2200$ vehs/hr from the capacity drop but otherwise performs similarly to the DRL policy.

Algorithm 3 Highway Bottleneck Derived Policy

procedure DERIVED(s, i) ▷ State s, AV index i
  $F \leftarrow$ get target inflow rate from s
  if $F \leq 2200$ **then**
    return Uncontrolled(s, i)
  Let j be the vehicle following i in the adjacent lane
  $x_1, x_2 \leftarrow$ tuned thresholds parameters
  $d_i, d_j \leftarrow$ distances to the merge point for i, j
  stop $\leftarrow$ j is uncontrolled **and** $d_i < x_1$ **and** $d_j < x_2$
  return $-c_{\text{decel}}$ **if** stop **else** c_{accel}

procedure UNCONTROLLED(s, i)
  return IDM acceleration for vehicle i based on s

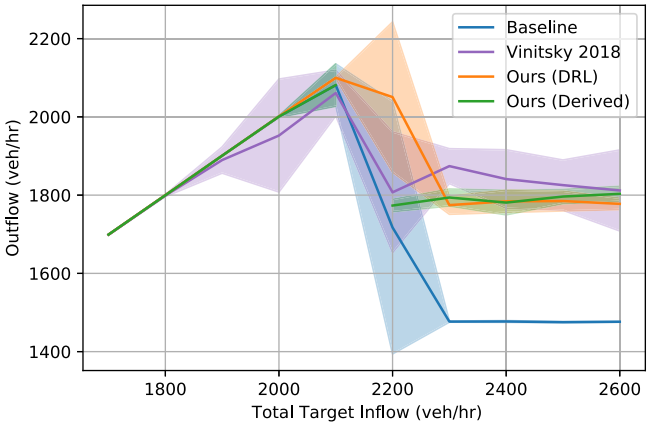


Fig. 10. **Highway Bottleneck Outflow.** We compare the outflow over horizon H under the Baseline, DRL, and Derived policies, with the shaped deep reinforcement learning method Vinitsky 2018 [5] as additional comparison. Our DRL policy sees similar performance to Derived under most target inflow rates, though the former learns to mitigate the transition region ($F = 2200$ vehs/hr) better than the latter. Both policies are significantly better than Baseline at high target inflow rates. We visualize Derived as a piecewise function because Derived reverts to Baseline for $F \leq 2100$ vehs/hr and the optimal threshold parameters x_1, x_2 are shared for all $F \geq 2200$ vehs/hr. With better performance for $F \leq 2200$ and worse performance for $F \geq 2300$, our DRL policy performs similarly on average to Vinitsky 2018, which artificially restricts control of AVs to segments of the traffic system to encourage ramp metering-like behavior.

E. Highway Ramp

In the Highway Ramp system under the Baseline policy, the ramp vehicles merging onto the highway force the highway vehicles to slow down, causing stop-and-go waves to propagate backward along the highway. The DRL policy learns to control AVs to hold highway vehicles back (Figure 11) to allow merging at a higher speed (Figure 12). The traffic system is similar to the one studied in [14], though we directly use the outflow as the objective while the prior work designs a reward function to encourage the speed of highway vehicle towards a manually specified v_{des}.

Observing the AV behavior under the DRL policy, we construct the Derived policy to similarly hold back highway vehicles distant from the merge point towards a tuned speed parameter v_{target} to allow for higher speed at the merge point. If the highway ahead is congested, v_{target} is temporarily set to 0 to allow congestion to ease. The Derived policy performs similarly to the DRL policy but requires more information on the congestion in front of the AV, which is provided as n_{leaders}.

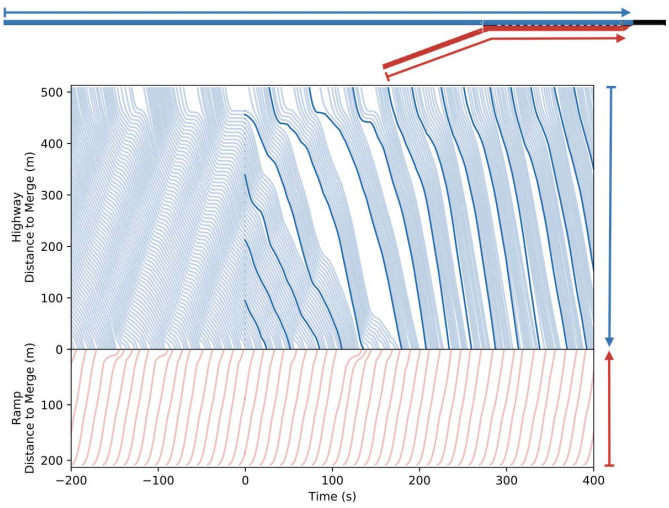


Fig. 11. **Highway Ramp time-space diagrams.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Bold indicates the AVs controlled by the DRL policy. Colored arrows indicate progressions of highway and ramp vehicles approaching the merge. While vehicles slow down at the merge point in Baseline, DRL learns to regulate the upstream speed of the highway vehicles so that vehicles at the merge point do not slow down.

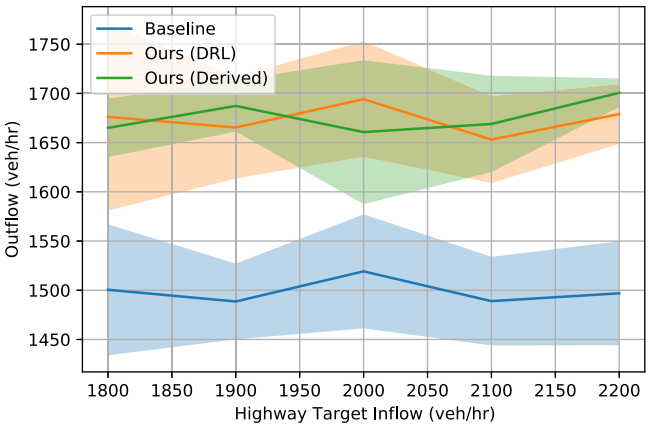


Fig. 12. **Highway ramp outflow.** We compare the outflow over horizon H under the Baseline, DRL, and Derived policies. Derived and DRL perform similarly for all target inflow rates; unlike the Derived policy, the DRL policy is not informed of the congestion ahead of each AV and faces a more difficult task. We display Derived as a single curve because the same target speed parameter v_{target} is optimal for all F considered.

F. Intersection

The Baseline Intersection system suffers severely from vehicles alternating to pass the intersection. DRL-controlled AVs not only learn to alternate less frequently, but they also learn to synchronize with AVs on opposite lanes (Figure 13). These learned behaviors resemble those of an adaptive traffic signal, greatly improving intersection throughput over the Baseline policy (Figure 14). Therefore, we design the Derived policy to follow a traffic signal-like behavior parameterized

Algorithm 4 Highway Ramp Derived Policy

procedure DERIVED(s, i) ▷ State s, AV index i
  $d_i \leftarrow$ distance to the merge point for AV i
  if $d_i \leq 400$ **then**
    return Uncontrolled(s, i)
  $v_{\text{target}} \leftarrow$ tuned speed parameter
  $v_i \leftarrow$ speed of AV i
  $n_{\text{leaders}} \leftarrow$ number of vehicles in front of i
  if $n_{\text{leaders}} > 20$ **then** ▷ Congested ahead
    $v_{\text{target}} \leftarrow 0$ ▷ Wait for congestion to clear
  return Equalize(v_{target}, v_i)

Algorithm 5 Intersection Derived Policy

procedure DERIVED(s, i) ▷ State s, AV index i
  $\ell_i, d_i \leftarrow$ lane, distance to intersection of AV i
  if $d_i \geq 15$ **then**
    return Uncontrolled(s, i)
  $t_H, t_V \leftarrow$ tuned phase parameters
  $t \leftarrow$ current simulation step $\mod (t_H + t_V)$
  phase $\leftarrow$ horizontal **if** $t < t_H$ **else** vertical
  if ℓ_i does not match phase **then**
    return $-c_{\text{decel}}$
  else if uncontrolled vehicles are crossing **then**
    return $-c_{\text{decel}}$
  else return c_{accel}

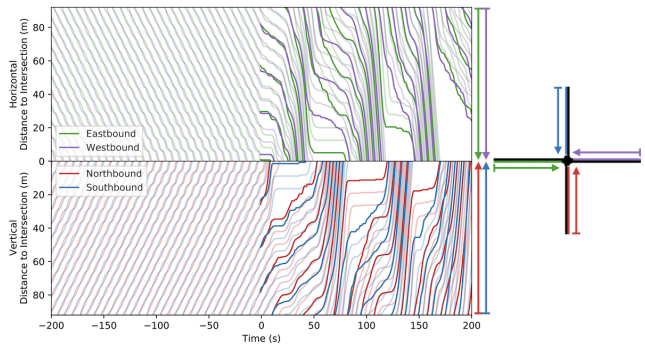


Fig. 13. **Intersection time-space diagrams.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Bold indicates the AVs controlled by the DRL policy. Colored arrows indicate progressions of vehicles on all lanes approaching the intersection. We see that the DRL policy develops an efficient traffic-signal-like behavior for grouping multiple vehicles and synchronizing the opposite lanes, whereas vehicles sees a stop-sign-like behavior under the Baseline policy.

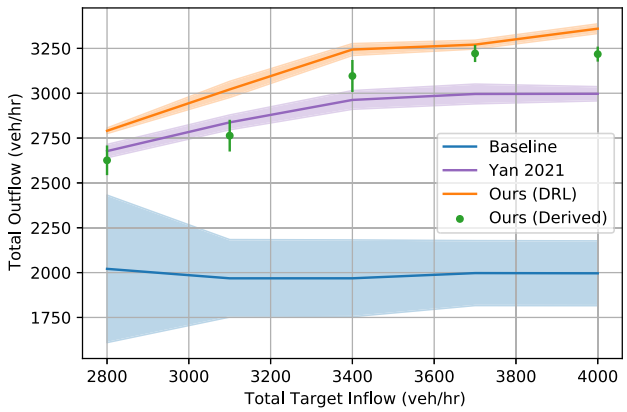


Fig. 14. **Intersection outflow.** We compare the outflow over horizon H under the Baseline, DRL, and Derived policies, with the deep reinforcement learning method Yan 2021 [17] as additional comparison. Note that performance is measured on all combinations of $(F_H, F_V) \in \{400, 550, 700, 850, 1000\}$ vehs/hr such that the total target inflow rate $F = 2(F_H + F_V)$ satisfies 2800 vehs/hr $\leq F \leq$ 4000 vehs/hr. Though Derived attempts to mimic the traffic signal behavior of our DRL policy with tuned horizontal and vertical phases, we find it difficult to achieve DRL-level performance with handcrafting. This suggests that the DRL controller is both performant and robust across configurations of F_H and F_V. Our DRL policy significantly outperforms Yan 2021 for all densities of traffic.

by horizontal and vertical phase t_H and t_V, which are tuned for each density configuration, with no yellow time. The AV additionally yields to any uncontrolled vehicles currently crossing the intersection. Though t_H and t_V are tuned independently for each configuration, we find that the Derived policy suffers from occasional lapses into alternation. In an additional comparison to Yan 2021 [17], we demonstrate that our present learning rate-free TRPO-based methodology offers significant advantages over a REINFORCE-based methodology, which obtains worse performance even with careful tuning of the learning rate.

VIII. CONCLUSION

This article introduces a unified and straightforward methodology for optimizing vehicular systems with mixed or full autonomy. While we demonstrate the generality and effectiveness of our methodology on several mixed autonomy traffic systems, the same methodology could be adapted to other vehicular robotic systems [3]. While our previous works applying DRL to mixed autonomy traffic often require extensive hyperparameter tuning and reward shaping, we show that the methodology presented in this work requires minimal hand-design and hyperparameter tuning. The performance and robustness of trained policies are characterized by comparisons with tuned rule-based policies. Finally, we provide future researchers and practitioners a lightweight framework which may be easily adapted to other systems and domains. While our earlier works based on the Flow framework [4], [13]–[16] were restricted by reliance on the general-use and heavyweight RLLib library [60], the unified methodology presented in this work is the result of our new lightweight framework which allowed more flexible research into the efficacy of various methodological components.

Further Sim2Real research and engineering are likely required for deployment in physical systems. In particular, future research may inject additional randomization and stochasticity in various aspects of the simulation to facilitate the learning of robust policies for Sim2Real transfer, while the design of the real-world system could be adjusted to reduce modeling error as much as possible. We argue that near-future Sim2Real extensions of our work are feasible for automated systems with existence of high fidelity simulators and safety mechanisms, as long as human intent does not need to be simulated; these criteria are likely satisfied already in industrial robotic settings. As real-world deployment may benefit

greatly from interpretability of trained policies, an impactful direction of future research may design an automatic method for distillation of trained DRL policies into Derived policies with interpretable behavioral building blocks, which could be shared across multiple systems. For example, convergence to fixed speed is a behavioral building block shared across DRL and Derived policies in Single Ring, Double Ring, and Figure Eight; waiting for a desired condition is another behavioral building block shared across Figure Eight, Highway Bottleneck, Highway Ramp, and Intersection.

Open directions of research in vehicular systems and traffic systems include 1) application of our methodology to vehicular systems beyond traffic systems, 2) application to richer traffic systems with other maneuvers such as turning and other control elements such as traffic signals, 3) optimizing for non-efficiency objectives, such as fuel or comfort, 4) optimizing the behavior of heterogeneous vehicles with different physical properties, 5) systems with non-stationary traffic regimes (*e.g.* natural variation of inflow rates), and 6) scaling up to larger systems by leveraging decomposition techniques. Due to multi-task training over many density configurations, we believe that our methodology already naturally handles non-stationary traffic regimes in particular, while the other open directions of research require further investigation.

APPENDIX
ROBUSTNESS OF DERIVED POLICIES UNDER RANGES OF CAR FOLLOWING MODEL PARAMETERS

As uncontrolled vehicles in our simulated traffic systems follow the IDM [48] car following model, which models human driving with a set of behavioral parameters, simulation dynamics may differ under differing IDM parameters. To probe the robustness of our Derived policies in the context of other car following parameters, we consider the Highway Bottleneck at a total target inflow of $F = 2600$ vehs/hr. Here, we identify the default IDM parameters as maximum acceleration $a = 2.6$ m/s^2, comfortable deceleration $b = 4.5$ m/s^2, desired velocity $v_0 = 30$ m/s, minimum spacing $s_0 = 2.5$ m, desired time headway $\tau = 1$ s, and exponent $\delta = 4$. In Table I, we document the effect of reasonable changes in IDM parameters on the performance of the Derived and Baseline policies from Section VII-D, holding the parameter of the Derived policy constant.

Derived outperforms the Baseline for all parameter combinations, and the performance of both policies are positively correlated with δ and v_0 and negatively correlated with τ and s_0. The performance of Derived is positively correlated with (a, b) while that of the Baseline is uncorrelated. In general, IDM parameter values modeling aggressive driving tend to improve the performance relative to those modeling conservative driving. Larger acceleration and deceleration parameters (a, b), higher desired velocity v_0, smaller minimum spacing s_0 between vehicles, and smaller desired time headway τ all intuitively correspond to more aggressive driving, while smaller exponent δ increases the aggressiveness of accelerations when the vehicle speed is near v_0. The only exception is the lack of correlation between Baseline performance and reasonable ranges of (a, b), suggesting that the inefficient alternation of vehicles at the bottleneck is not due to insufficient maximum acceleration a.

TABLE I
HIGHWAY BOTTLENECK OUTFLOW (VEHS/HR) UNDER DIFFERENT IDM PARAMETERS AT TOTAL TARGET INFLOW $F = 2600$ M/S. EACH PARAMETER TABLE HOLDS ALL OTHER PARAMETERS AT THE DEFAULT VALUE. IN ALL CASES, MEANS AND STANDARD DEVIATIONS ARE COMPUTED OVER 10 INDEPENDENTLY SAMPLED TRAJECTORIES

(a, b)	$(1, 1.5)$	$(2, 3)$	$(2.6, 4.5)$ (Default)
Baseline	1476 ± 3	1476 ± 2	1476 ± 2
Derived	1619 ± 27	1779 ± 24	1787 ± 26

τ	0.5	0.75	1 (Default)	1.25
Baseline	1696 ± 222	1540 ± 1	1476 ± 2	1456 ± 3
Derived	2040 ± 30	1953 ± 19	1787 ± 26	1632 ± 15

v_0	15	20	25	30 (Default)
Baseline	1463 ± 3	1474 ± 2	1475 ± 3	1476 ± 2
Derived	1679 ± 13	1750 ± 31	1782 ± 20	1787 ± 26

s_0	2	2.5 (Default)	3
Baseline	1514 ± 2	1476 ± 2	1440 ± 1
Derived	1843 ± 28	1787 ± 26	1736 ± 17

δ	2	3	4 (Default)	5
Baseline	1458 ± 2	1473 ± 2	1476 ± 2	1478 ± 2
Derived	1724 ± 17	1773 ± 26	1787 ± 26	1791 ± 24

ACKNOWLEDGMENT

The authors acknowledge MIT SuperCloud and the Lincoln Laboratory Supercomputing Center for providing computational resources supporting the research results in this paper. The authors are grateful for the constructive suggestions by all reviewers and editors.

REFERENCES

[1] K. C. Morris, C. Schlenoff, and V. Srinivasan, "Guest editorial a remarkable resurgence of artificial intelligence and its impact on automation and autonomy," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 407–409, Apr. 2017.

[2] Z. Wadud, D. MacKenzie, and P. Leiby, "Help or hindrance? The travel, energy and carbon impacts of highly automated vehicles," *Transp. Res. A, Policy Pract.*, vol. 86, pp. 1–18, Apr. 2016.

[3] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Mag.*, vol. 29, no. 1, p. 9, 2008.

[4] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1270–1286, Apr. 2022.

[5] E. Vinitsky, K. Parvate, A. Kreidieh, C. Wu, and A. Bayen, "Lagrangian control through deep-RL: Applications to bottleneck decongestion," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 759–765.

[6] K. Jang *et al.*, "Simulation to scaled city: Zero-shot policy transfer for traffic control via autonomous vehicles," in *Proc. 10th ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, Apr. 2019, pp. 291–300.

[7] S. Hofer *et al.*, "Sim2Real in robotics and automation: Applications and challenges," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 398–400, Apr. 2021.

[8] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Ensemble coordination approach in multi-AGV systems applied to industrial warehouses," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 922–934, Jul. 2015.

[9] W.-S. Lin and J.-W. Sheu, "Optimization of train regulation and energy usage of metro lines using an adaptive-optimal-control algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 855–864, Oct. 2011.

[10] R. Morris *et al.*, "Planning, scheduling and monitoring for airport surface operations," in *Proc. Workshops 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.

[11] D. Stavrou, S. Timotheou, C. G. Panayiotou, and M. M. Polycarpou, "Optimizing container loading with autonomous robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 717–731, Apr. 2018.

[12] Y. Zhu, D. Zhao, and H. He, "Optimal feedback control of pedestrian flow in heterogeneous corridors," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1097–1108, Jul. 2021.

[13] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Emergent behaviors in mixed-autonomy traffic," in *Proc. Conf. Robot Learn.*, 2017, pp. 398–407.

[14] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1475–1480.

[15] E. Vinitsky *et al.*, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Proc. Conf. Robot Learn.*, 2018, pp. 399–409.

[16] E. Vinitsky, N. Lichtle, K. Parvate, and A. Bayen, "Optimizing mixed autonomy traffic flow with decentralized autonomous vehicles and multi-agent RL," 2020, *arXiv:2011.00120*.

[17] Z. Yan and C. Wu, "Reinforcement learning for mixed autonomy intersections," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2089–2094.

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

[19] J. Little, M. Kelson, and N. Gartner, "MAXBAND: A program for setting signals on arteries and triangular networks," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 795, pp. 40–46, Dec. 1981.

[20] P. Hunt, D. Robertson, R. Bretherton, R. Winton, Transport, and R. R. Laboratory, *SCOOT: A Traffic Responsive Method Coordinating Signals* (TRRL Laboratory Report). Crowthorne, U.K.: Transport and Road Research Lab, 1981.

[21] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin, Germany: Springer, 2013.

[22] M. Papageorgiou, H. Hadj-Salem, and J. M. Blosseville, "ALINEA: A local feedback control law for on-ramp metering," *Transp. Res. Rec.*, vol. 1320, no. 1, pp. 58–67, 1991.

[23] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 429–436, Dec. 2006.

[24] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Cham, Switzerland: Springer, 2009.

[25] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 3, pp. 143–153, Sep. 2003.

[26] S. Lefèvre, A. Carvalho, and F. Borrelli, "A learning-based framework for velocity control in autonomous driving," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 32–42, Jan. 2016.

[27] G. Sharon and P. Stone, "A protocol for mixed autonomous and human-operated vehicles at intersections," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, Cham, Switzerland: Springer, 2017, pp. 151–167.

[28] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.

[29] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: An ant colony system for autonomous intersection management," *Appl. Intell.*, vol. 37, no. 2, pp. 207–222, 2012.

[30] D. Miculescu and S. Karaman, "Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 680–694, Feb. 2020.

[31] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020.

[32] E. Ko and K.-C. Chen, "Wireless communications meets artificial intelligence: An illustration by autonomous vehicles on Manhattan streets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[33] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 1996.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[35] H. Wei *et al.*, "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1913–1922.

[36] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1198–1207, Apr. 2018.

[37] Z. Liu *et al.*, "Visuomotor reinforcement learning for multirobot cooperative navigation," *IEEE Trans. Autom. Sci. Eng.*, early access, Oct. 1, 2021, doi: 10.1109/TASE.2021.3114327.

[38] Q. Xiao, C. Li, Y. Tang, and L. Li, "Meta-reinforcement learning of machining parameters for energy-efficient process control of flexible turning operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 5–18, Jan. 2021.

[39] I.-B. Park, J. Huh, J. Kim, and J. Park, "A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1420–1431, Jul. 2020.

[40] X. Ou, Q. Chang, and N. Chakraborty, "A method integrating Q-learning with approximate dynamic programming for gantry work cell scheduling," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 85–93, Jan. 2021.

[41] C. Wu *et al.*, "Framework for control and deep reinforcement learning in traffic," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[42] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[43] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[44] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[46] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited*, vol. 14, no. 8, p. 2, 2012.

[47] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–14.

[48] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, p. 1805, Aug. 2000.

[49] Z. Liu, H. Wang, H. Wei, M. Liu, and Y.-H. Liu, "Prediction, planning, and coordination of thousand-warehousing-robot networks with motion and communication uncertainties," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1705–1717, Oct. 2021.

[50] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, nos. 1–2, pp. 99–134, 1998.

[51] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Proc. TARK*, vol. 96. Princeton, NJ, USA: Citeseer, 1996, pp. 195–210.

[52] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, Cham, Switzerland: Springer, 2017, pp. 66–83.

[53] L. Engstrom *et al.*, "Implementation matters in deep RL: A case study on PPO and TRPO," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[54] S. M. Kakade, "A natural policy gradient," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001, pp. 1–8.

[55] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *J. Mach. Learn. Res.*, vol. 22, no. 98, pp. 1–76, 2021.

[56] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.

[57] R. Herman, E. W. Montroll, R. B. Potts, and R. W. Rothery, "Traffic dynamics: Analysis of stability in car following," *Oper. Res.*, vol. 7, no. 1, pp. 86–106, 1959.

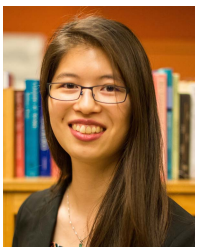
[58] R. E. Stern *et al.*, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 205–221, Apr. 2018.
[59] M. Saberi and H. Mahmassani, "Hysteresis and capacity drop phenomena in freeway networks: Empirical characterization and interpretation," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2391, pp. 44–55, Dec. 2013.
[60] E. Liang *et al.*, "Rllib: Abstractions for distributed reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3053–3062.

Zhongxia Yan (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering and computer science from the University of California, Berkeley, Berkeley, CA, USA, in 2017 and 2018, respectively. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with the Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include application of machine learning and reinforcement learning to problems in transportation and logistics. He was a recipient of the Department of Transportation DDETFP Graduate Fellowship.

Abdul Rahman Kreidieh (Member, IEEE) received the B.S. degree in mechanical engineering from the American University of Beirut, Beirut, Lebanon, in 2016, and the M.S. degree in civil and environmental engineering from the University of California, Berkeley, Berkeley, CA, USA, in 2017, where he is currently pursuing the Ph.D. degree in civil and environmental engineering. His primary focus is on designing models and algorithms that scale the performance of existing machine learning systems to large-scale traffic control problems. His research interests include the intersection of machine learning and traffic control through mixed autonomy systems.

Eugene Vinitsky (Member, IEEE) received the B.S. degree in physics from the California Institute of Technology, Pasadena, CA, USA, in 2014, and the M.A. degree in physics from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2015. He is currently pursuing the Ph.D. degree in controls engineering with the Mobile Sensing Laboratory, University of California, Berkeley, Berkeley, CA, USA. His current research interests include multiagent reinforcement learning, cooperative automated vehicles, and robust control. He was a recipient of the National Science Foundation Graduate Research Fellowship.

Alexandre M. Bayen (Senior Member, IEEE) received the engineering degree in applied mathematics from the École Polytechnique, Palaiseau, France, in 1998, and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 1999 and 2004, respectively. He is the Liao-Cho Professor of Engineering with the University of California, Berkeley, Berkeley, CA, USA. He is currently a Professor of Electrical Engineering and Computer Science, and Civil and Environmental Engineering. He is currently the Director of the Institute of Transportation Studies. He is also a Faculty Scientist in Mechanical Engineering with the Lawrence Berkeley National Laboratory.



Cathy Wu (Member, IEEE) received the B.S. and M.Eng. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2012 and 2013, respectively, and the Ph.D. degree from the University of California, Berkeley, Berkeley, CA, USA, in 2018, all in electrical engineering and computer sciences. She was a Post-Doctoral Researcher with Microsoft Research AI. She is an Assistant Professor with MIT in LIDS, CEE, and IDSS. She studies the technical challenges surrounding the integration of autonomy into societal systems. Her research interests include machine learning and mobility. She was a recipient of several awards, including the 2019 IEEE Intelligent Transportation Systems Society (ITSC) Best Ph.D. Dissertation Award, the 2018 Milton Pikarsky Memorial Dissertation Award, and the 2016 IEEE ITSC Best Paper Award; and has appeared in the press, including Wired and Science.