# Unified Automatic Control of Vehicular Systems with Reinforcement Learning

Zhongxia Yan[1], Abdul Rahman Kreidieh[2], Eugene Vinitsky[2], Alexandre M. Bayen[2], Cathy Wu[1]

***Abstract*— Emerging vehicular systems with increasing proportions of automated components present opportunities for optimal control to mitigate congestion and increase efficiency. There has been 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.**

## 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. This trend can be observed in systems ranging from real-world traffic systems to warehouses employing mobile robots for storage, sorting, or delivery. 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.

Typically, 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 such systems, simulation decouples modeling of the system from further analysis and optimization. Thus, simulations of varying fidelity exist 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 DRL is applicable to both continuous and discrete domains and only requires the ability to simulate forward trajectories from a set of initial states. This work follows a series of our previous works applying DRL to mixed autonomy traffic [1]–[6]. While each previous work often focuses analyses on a single traffic system and applies significant amounts of system-specific handcrafting, this work presents a simplified and unified DRL methodology for a superset of open and closed traffic systems, with a focus on generality and ease of applicability. The code introduced in this work is a lightweight revision of the Flow Framework [1]. Additionally, we interpret the behaviors of DRL-controlled AVs, some of which resemble those designed by traffic engineering experts, and mimic the DRL policies with simple controllers.

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.

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

## II. Related Work

**Traffic control.** Due to the ubiquity and costs of congestion in traffic, much work has been devoted to traffic control for increasing local or system-wide efficiency. In urban traffic networks traffic signal control strategies have been widely studied for isolated or coordinated intersections [7]. In freeway traffic networks, ramp metering control methods like ALINEA [8] are deployed to manage reduction in road capacity. Studies of mixed and full autonomy control of freeway or intersections [9], [10] typically involve heuristic-based algorithms or simplified models. As discussed in more detail in [1], two prominent challenges in studying mixed autonomy in particular are the high uncertainty in system dynamics, due 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 addressing both challenges.

**Model-free DRL for mixed autonomy traffic.** This work generalizes our previous works on applications of model-free

[1] Laboratory for Information & Decision Systems, Massachusetts Institute of Technology {zxyan, cathywu}@mit.edu

[2] Mobile Sensing Lab, University of California, Berkeley {aboudy, evinitsky, bayen}@berkeley.edu

DRL to mixed autonomy [1]–[6], [11], [12]. While each previous work demonstrates that DRL overcomes long-standing classical control challenges in traffic control, these work often included artificial encouragement and handcrafting to guide the DRL policy in their specific traffic system. This work shows that a unified methodology achieves improved efficiency without resorting to system-specific hand-design to ease optimization. The ability to easily discover performant behaviors without hand-holding is one 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 $\rho_0$, discount factor $\gamma \in [0, 1]$, and horizon $H \in \mathbb{Z}_+$. Given this MDP definition, reinforcement learning and optimal control aim towards maximizing the expected cumulative reward by selecting optimal actions $a_0 \dots a_{H-1} \in \mathcal{A}$.

### 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, parameterized by $\theta$, is optimized to maximize the expected cumulative reward

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

The Trust Region Policy Optimization (TRPO) algorithm [13] collects a trajectory $(s_0, a_0 \dots, s_{H-1}, a_{H-1}, s_H)$ to optimize Equation 1. However, to encourage training stability, TRPO constrains the policy update:

$$\theta \leftarrow \arg\max_{\theta'} \sum_{t=0}^{H-1} \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \sum_{t'=t}^{H-1} \gamma^{t'-t} r(s_{t'}, a_{t'}, s_{t'+1})$$
$$\text{subject to } \frac{1}{H} \sum_{t=0}^{H-1} D_{\text{KL}}(\pi_\theta(\cdot|s) \| \pi_{\theta'}(\cdot|s)) \leq \delta_{\text{kl}}, \quad (2)$$

where $\delta_{\text{kl}}$ is the upper bound of the mean KL divergence between the updated policy $\pi_{\theta'}$ and the original policy $\pi_\theta$. We do not use or consider a critic in this work.

## IV. Automated Vehicular Systems

We describe the general 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.

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

### A. MDP Definition

We 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. $s_{t+1} \sim T(s_t, a_t, \cdot)$ can be sampled from the simulator, which applies the actions for all vehicles over a simulation step duration $\delta$. When the simulator safety checks are inadequate, we add a collision penalty to $r(s_t, a_t, s_{t+1})$.

### B. Partial Observability

In practice, as the state $s$ could be large or difficult to reason about, DRL methods often approximate the policy with $\pi_\theta(\cdot|s) \approx \pi_\theta(\cdot|o)$ [13], 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) [14].

### 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) [15]. 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 decomposition, the combinatorial nature of $\mathcal{A}$ poses an intractable problem to learning algorithms.

### D. Per-AV Action Space

We naturally formulate the longitudinal per-AV action space 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. 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 with 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 hidden size of 64. We share the policy parameter $\theta$ across all vehicles in the traffic network to share experiences between AVs [16]. 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. 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.

### G. 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. We permit Derived policies to use information from any part of the state, contrasting with DRL policies which must rely on observed information and generalize well across all density configurations.

## VI. Experimental Setup

### A. Vehicular Systems

We construct six diverse mixed autonomy traffic systems in the SUMO microscopic simulator [17] to demonstrate the generality of our unified methodology. Three systems are open and three systems are closed. We do not incorporate any traffic control element, such as traffic light or ramp meter. All vehicles are 5m in length and uncontrolled vehicles follow the Intelligent Driver Model (IDM) [18] with a Gaussian acceleration noise of 0.2m/s$^2$. 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. SUMO safety checks prevent vehicles from entering most collisions situations. We consider multiple traffic density configurations for each system.

In closed systems, the objective is 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, the objective is the throughput (outflow per hour) of the system. We use $\delta = 0.5$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.

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 1.

*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. 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

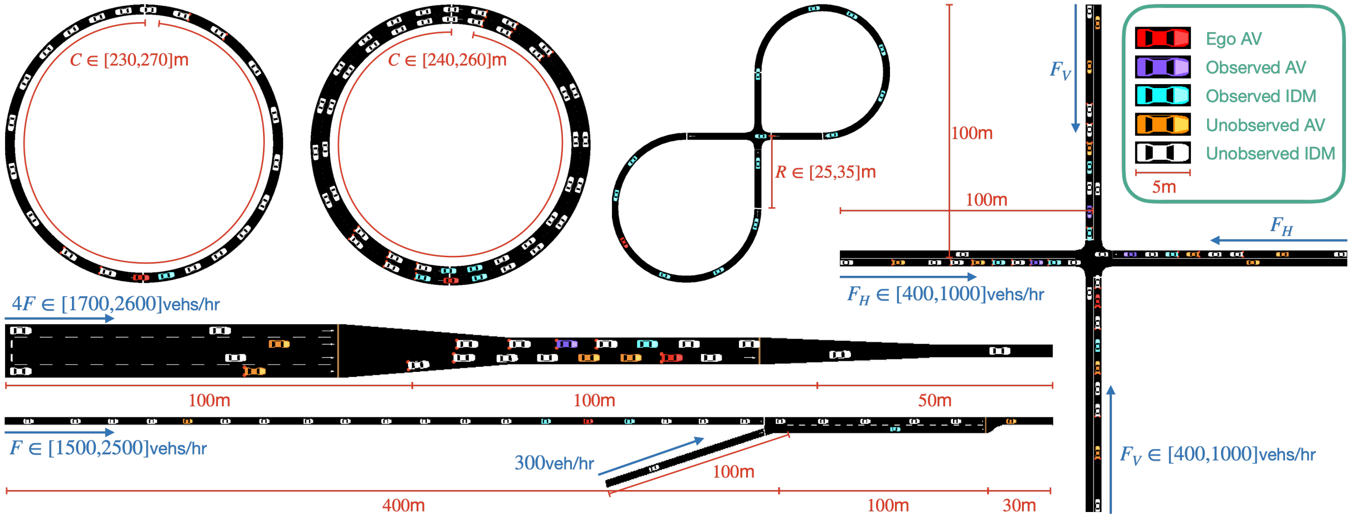


Fig. 1. **Experimental Traffic Systems.** In clockwise order from the top left: Single Ring, Double Ring, Figure Eight, Intersection, Highway Ramp, and Highway Bottleneck. Each traffic system is independently drawn to scale. Single Ring, Double Ring, and Figure Eight are closed systems with 22, 44, and 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.

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 $\{\text{Global}, \text{Greedy}\}$ 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. $r(s, a, s')$ 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 $F$; the total target inflow rate is $4F \in [1700, 2600]$ vehs/hr. At the first merge, 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 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 frequently occur in the Figure Eight and Intersection systems, 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 [1], [6], [12] to train policies within our respective traffic systems. 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.

### 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.

## VII. Experimental Results

We measure numerical performances after sufficient duration has passed for vehicle dynamics to achieve steady state. We compute the means and standard deviations across 10 trajectories with different seeds. We dissect and visualize the behavior via time-space diagrams for representative configurations of each traffic system studied.

### A. Single Ring

Due to the linear string instability of IDM [19], the Baseline policy quickly results in a stop-and-go waves which propagate in the opposite direction of traffic [20] 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 2.

Mimicking this behavior, we design a Derived policy with a single, optimized target speed $v_{\text{target}}$ per circumference configuration. Figure 3 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 [1] (Figure 3) 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, [1] only considers a global objective, while we consider both global and greedy.

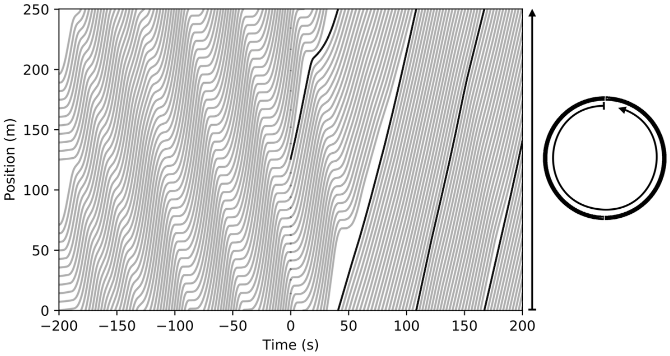


Fig. 2. **Single Ring** $C = 250$ m **Time-space Diagram.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned Global policy (on and after time 0s). Bold indicates the AV controlled by the DRL policy. Arrows indicate progression of vehicles. The DRL policy controls the AV to eliminate the backward propagating waves formed under the Baseline policy.

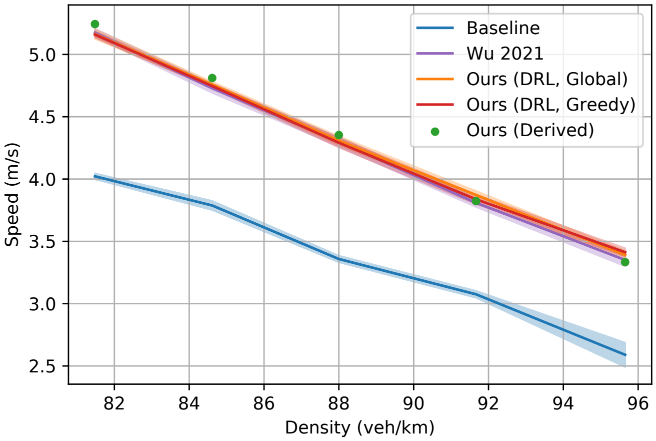


Fig. 3. **Single Ring Average Speed.** We compare the average speed over all 22 vehicles over horizon $H$ under the Baseline, DRL, and Derived policies, with the shaped deep reinforcement learning policy Wu 2021 [1] as additional comparison. Our DRL and Derived policies see significantly better average speeds than those of the Baseline policy. We display the Derived performance as points instead of a single line because the optimal speed parameter $v_{\text{target}}$ 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.

**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($v_{\text{target}}, v$)

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

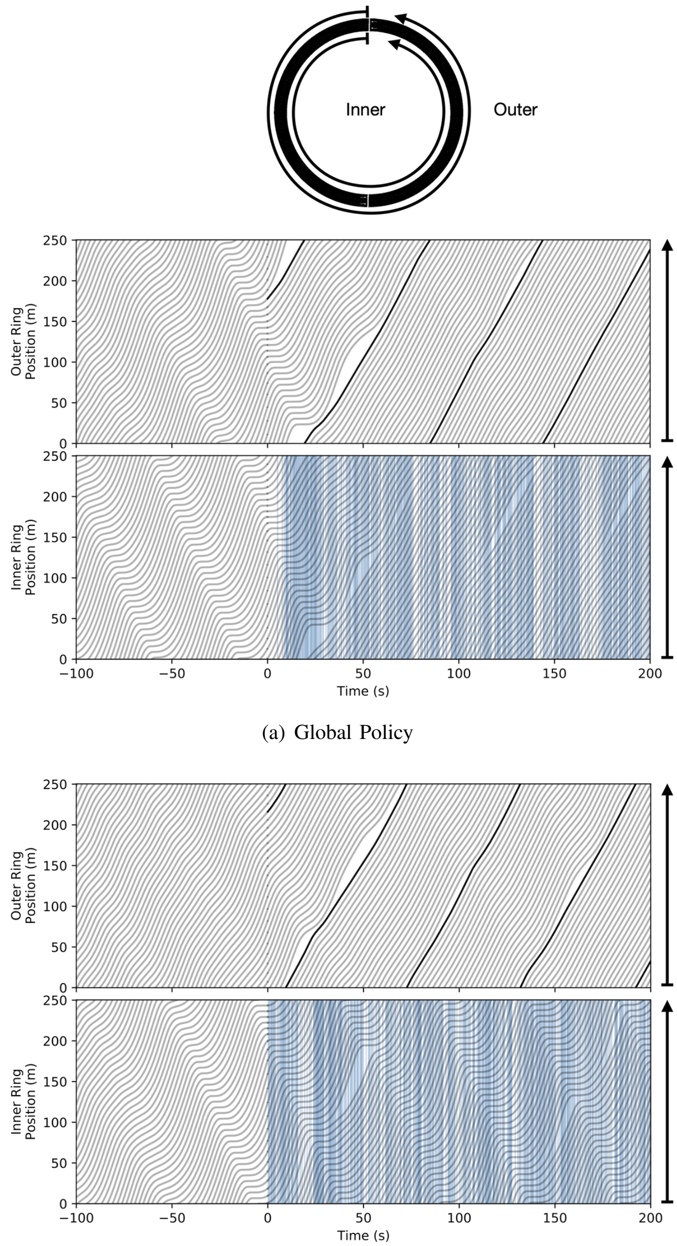
### *B. Double Ring*

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 4 and compared numerically in Figure 5. We note that the AV under the Greedy policy also frequently flickers its turn signal, as seen in Figure 4; 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 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; we are not able to construct 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 intersection one by one, similar to the behavior at a stop-sign. As shown in Figure 6, 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 7. Using the same approach as the Single Ring, we design the Derived policy by applying exhaustive search to find an optimal target speed $v_{\text{target}}$. We find that DRL achieves close to the tuned target speed for all configurations.

While [4] 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.



(a) Global Policy

(b) Greedy Policy

Fig. 4. **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.

### *D. Highway Bottleneck*

The Highway Bottleneck under the Baseline policy sees two distinct behaviors: at low target inflow rates $F <$

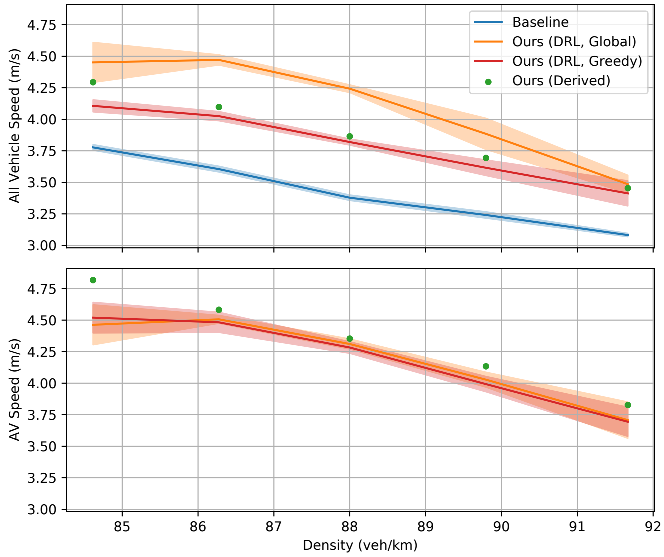


Fig. 5. **Double Ring Average and AV Speed.** Over horizon $H$, we compare the average speed over all 44 vehicles (top) and AV speed (bottom) under the Baseline, DRL, and Derived policies. The Global policy sees the best average speed in almost all cases, due to mitigation of stop-and-go waves in both 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.

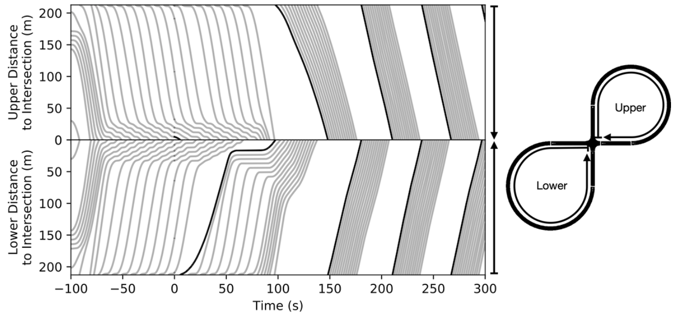


Fig. 6. **Figure Eight** $R = 30$ m **Time-space Diagram.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Bold indicates the AV controlled by the DRL policy. Arrows indicate progressions of vehicles approaching the intersection from the upper and lower loop. The AV guides a snaking behavior that eliminates alternation of single vehicles at the intersection.

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 [21] 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 8) 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 9).

We consider the DRL method introduced by Vinitsky 2018 [12] as an additional baseline in Figure 9. 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 [12] with the RMSprop optimizer [22], which we find to improve performance over the ADAM optimizer [23] used by [12]. 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 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.

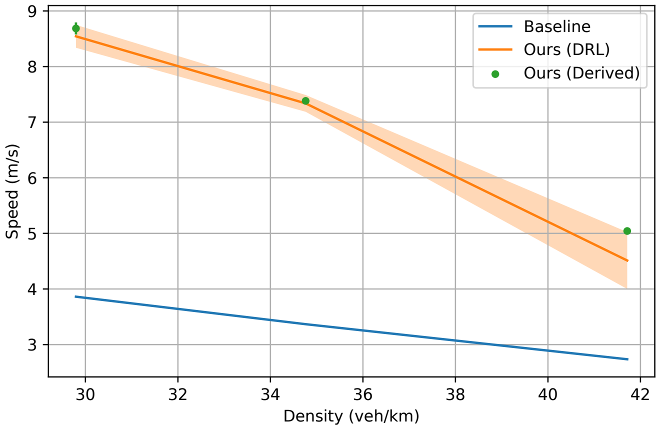
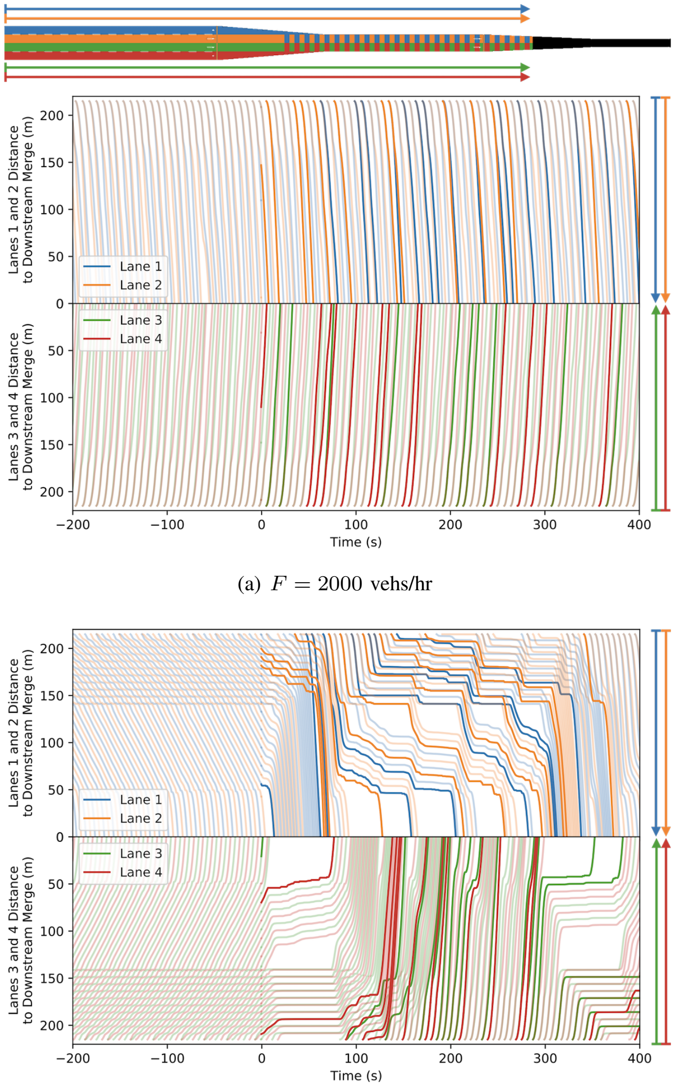


Fig. 7. **Figure Eight Average Speed.** We compare the average speed over all 14 vehicles over horizon $H$ under the Baseline, DRL, and Derived policies. The DRL policy nearly matches the Derived policy, despite needing to infer the target speed from solely the observations.

**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($v_{\text{target}}, v$)
```

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.



(a) $F = 2000$ vehs/hr

(b) $F = 2400$ vehs/hr

Fig. 8. **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.

### 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 10) to allow merging at a higher speed (Figure 11). The traffic system is similar to the one studied in [3], 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 $v_{\text{des}}$.

**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 $c_{\text{accel}}$

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

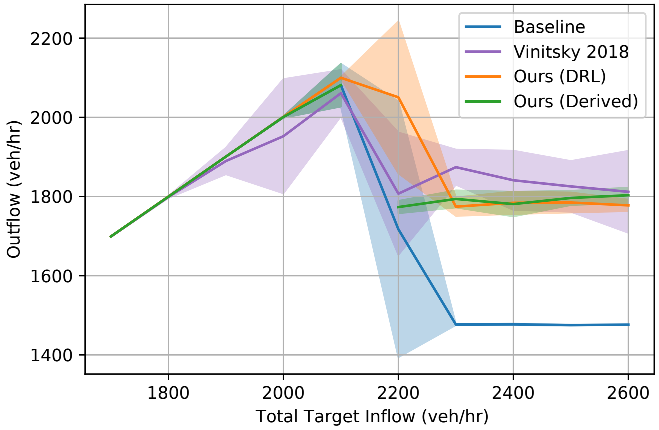


Fig. 9. **Highway Bottleneck Outflow.** We compare the outflow over horizon $H$ under the Baseline, DRL, and Derived policies, with the shaped deep reinforcement learning method Vinitsky 2018 [12] as additional comparison. Our DRL policy sees similar performance to Derived under most target inflow rates, 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.

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 $v_{\text{target}}$ to allow for higher speed at the merge point. If the highway ahead is congested, $v_{\text{target}}$ 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 $n_{\text{leaders}}$.

### 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 12).

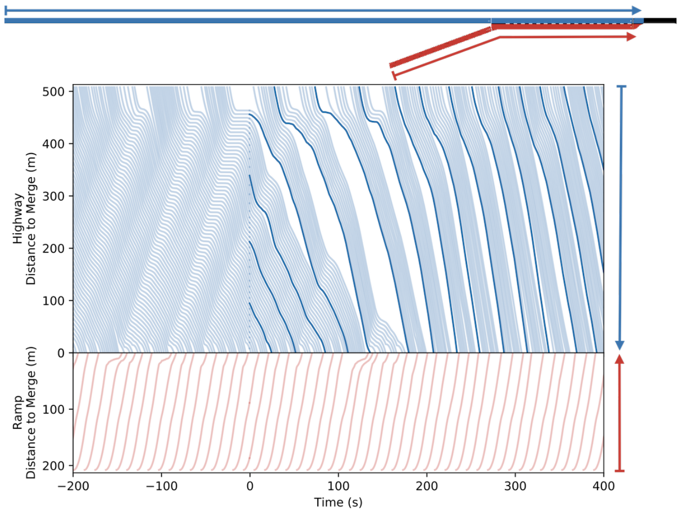


Fig. 10. **Highway Ramp Time-space Diagrams.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Bold indicates the AVs controlled by the DRL policy. Colored arrows indicate progressions of highway and ramp vehicles approaching 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.

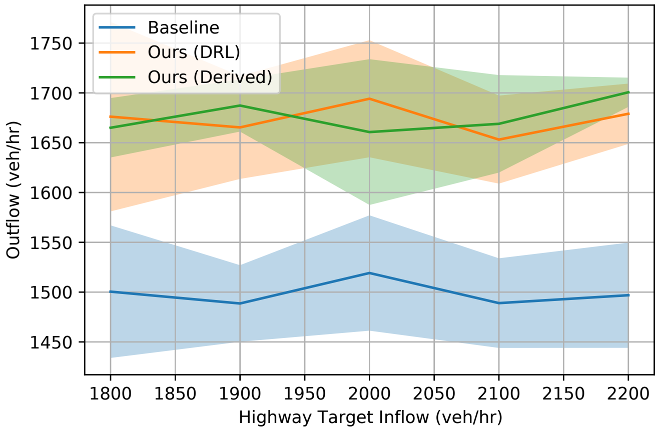


Fig. 11. **Highway Ramp Outflow.** We compare the outflow over horizon $H$ under the Baseline, DRL, and Derived policies. Derived and DRL perform similarly for all target inflow rates; unlike the Derived policy, the DRL policy is not informed of the congestion ahead of each AV and faces a more difficult task. We display Derived as a single curve because the same target speed parameter $v_{\text{target}}$ is optimal for all $F$ considered.

These learned behaviors resemble those of an adaptive traffic signal, greatly improving intersection throughput over the Baseline policy (Figure 13). Therefore, we design the Derived policy to follow a traffic signal-like behavior parameterized 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 [6], 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.

**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($v_{\text{target}}, v_i$)
```

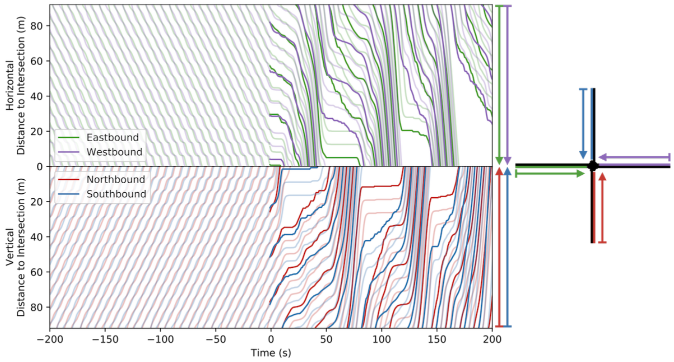


Fig. 12. **Intersection Time-space Diagrams.** We plot the trajectories of vehicles under the Baseline policy (before time 0s) and the learned DRL policy (on and after time 0s). Bold indicates the AVs controlled by the DRL policy. Colored arrows indicate progressions of vehicles on all lanes approaching 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.

## 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 [24]. 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. 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.

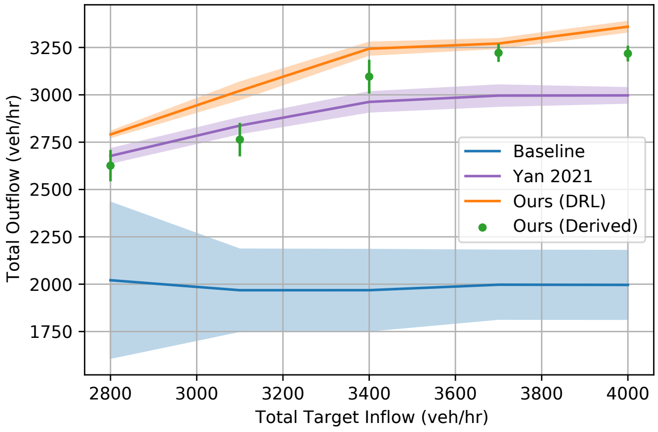


Fig. 13. **Intersection Outflow.** We compare the outflow over horizon $H$ under the Baseline, DRL, and Derived policies, with the deep reinforcement learning method Yan 2021 [6] as additional comparison. Note that performance is measured on all combinations of $(F_H, F_V) \in \{400, 550, 700, 850, 1000\}$ 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.

**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 $\ell_i$ does not match phase then
        return $-c_{\text{decel}}$
    else if uncontrolled vehicles are crossing then
        return $-c_{\text{decel}}$
    else return $c_{\text{accel}}$
```

## Acknowledgment

We MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing computational resources. This research was supported by Amazon, MIT-IBM Watson AI Lab, and the DDETFP Fellowship Program.

## References

[1] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Transactions on Robotics*, 2021.

[2] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen, "Emergent behaviors in mixed-autonomy traffic," in *Conference on Robot Learning*. PMLR, 2017, pp. 398–407.

[3] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1475–1480.

[4] E. Vinitsky, A. Kreidieh, L. Le Flem, N. Kheterpal, K. Jang, C. Wu, F. Wu, R. Liaw, E. Liang, and A. M. Bayen, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Conference on robot learning*. PMLR, 2018, pp. 399–409.

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

[6] Z. Yan and C. Wu, "Reinforcement learning for mixed autonomy intersections," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2089–2094.

[7] P. Hunt, D. Robertson, R. Bretherton, R. Winton, Transport, and R. R. Laboratory, *SCOOT: A Traffic Responsive Method of Coordinating Signals*, ser. TRRL Laboratory report. TRRL Urban Networks Division, 1981.

[8] M. Papageorgiou, H. Hadj-Salem, J.-M. Blosseville *et al.*, "Alinea: A local feedback control law for on-ramp metering," *Transportation research record*, vol. 1320, no. 1, pp. 58–67, 1991.

[9] B. Van Arem, C. J. Van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 4, pp. 429–436, 2006.

[10] D. Miculescu and S. Karaman, "Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 680–694, 2019.

[11] C. Wu, K. Parvate, N. Kheterpal, L. Dickstein, A. Mehta, E. Vinitsky, and A. M. Bayen, "Framework for control and deep reinforcement learning in traffic," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–8.

[12] E. Vinitsky, K. Parvate, A. Kreidieh, C. Wu, and A. Bayen, "Lagrangian control through deep-rl: Applications to bottleneck decongestion," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 759–765.

[13] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[14] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[15] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *TARK*, vol. 96. Citeseer, 1996, pp. 195–210.

[16] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 2017, pp. 66–83.

[17] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

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

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

[20] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli *et al.*, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205–221, 2018.

[21] M. Saberi and H. S. Mahmassani, "Hysteresis and capacity drop phenomena in freeway networks: empirical characterization and interpretation," *Transportation research record*, vol. 2391, no. 1, pp. 44–55, 2013.

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

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

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