In most modern cities, traffic congestion is one of the most salient societal challenges. Past research has shown that inserting a limited number of autonomous vehicles (AVs) within the traffic flow, with driving policies learned specifically for the purpose of reducing congestion, can significantly improve traffic conditions. However, to date these AV policies have generally been evaluated under the same limited conditions under which they were trained. On the other hand, to be considered for practical deployment, they must be robust to a wide variety of traffic conditions. This article establishes for the first time that a multiagent driving policy can be trained in such a way that it generalizes to different traffic flows, AV penetration, and road geometries, including on multi-lane roads. Inspired by our successful results in a high-fidelity microsimulation, this article further contributes a novel extension of the well-known Cell Transmission Model (CTM) that, unlike past CTMs, is suitable for modeling congestion in traffic networks, and is thus suitable for studying congestion-reduction policies such as those considered in this article.

Learning a Robust Multiagent Driving Policy for Traffic Congestion Reduction

Yulin Zhang[1,2*], William Macke[2], Jiaxun Cui[2], Sharon Hornstein[3], Daniel Urieli[3] and Peter Stone[2,4]

[1*]Amazon Robotics, 300 Riverpark Dr., North Reading, 01864, Massachusetts, United States. The work was done prior to joining Amazon.
[2]Department of Computer Science, The University of Texas at Austin, 2317 Speedway, Austin, 78712, Texas, United States.
[3]General Motors Israel R&D Labs.
[4]Sony AI.

*Corresponding author(s). E-mail(s): zhangyl@amazon.com;
Contributing authors: wmacke@cs.utexas.edu; cuijiaxun@utexas.edu;
sharon.hornstein@gm.com; daniel.urieli@gm.com; pstone@cs.utexas.edu;

Keywords: Autonomous Vehicles, Deep Reinforcement Learning, Traffic Optimization, Multiagent Systems, Multiagent Reinforcement Learning, Flow

1 Introduction

According to Texas A&M's 2021 Urban Mobility Report, traffic congestion in 2020 in the U.S. was responsible for excess fuel consumption of about 1.7 billion gallons, an annual delay of 4.3 billion hours, and a total cost of $100B [1]. A common form of traffic congestion on highways is *stop-and-go waves*, which have been shown in field experiments to emerge when vehicle density exceeds a critical value [2]. Past research has shown that in human-driven traffic, a small fraction of automated or autonomous vehicles (AVs) executing a controlled multiagent driving policy can mitigate stop-and-go waves in simulated and real-world scenarios, roughly double the traffic speed, and increase throughput by about 16% [3]. Frequently, the highest-performing policies are those learned by deep reinforcement learning (DRL) algorithms, rather than hand-coded or model-based driving policies.

Any congestion reduction policy executed in the real world will need to perform robustly under a wide variety of traffic conditions such as traffic flow, AV penetration (percentage of AVs in traffic, referred to

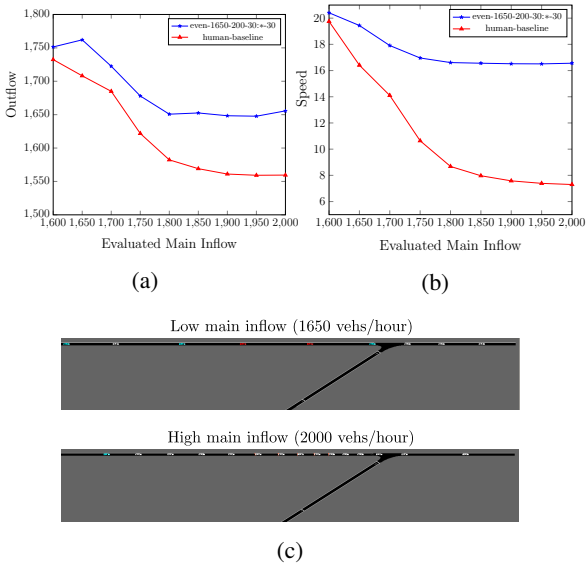


Fig. 1: Increasing incoming vehicle flow (the demanded *inflow*) degrades performance of a policy trained with inflow of 1650 veh/hour, with respect to both throughput (a) and speed (b). A visual representation (c) is given that shows what this decreased efficiency looks like. The red curve shows the performance of a human baseline with no AVs (AVP=0), and the blue curve shows the performance of a trained policy with 30% AVs (AVP=30).

here as "AVP"), AV placement in traffic, and road geometry. However, existing driving policies have generally been tested in the same conditions they were trained on, and have not been thoroughly tested for robustness to different traffic conditions. Indeed, their performance can degrade considerably when evaluated outside of the training conditions (Figure 1). Therefore, it remains unclear how to create a robust DRL congestion-reduction driving policy that is practical for real-world deployment.

In this article, we establish for the first time the existence of a robust DRL congestion-reduction driving policy that performs well across a wide variety of traffic flows, AVP, AV placement in traffic, and several road geometries. Moreover, we investigate the question of how to come up with such a policy and what degree of robustness it can achieve. We create a testbed with a diverse, pre-defined collection of test traffic conditions of real-world interest including the single-lane merge scenario shown in Figure 1c. Such merge scenarios are a common source of stop-and-go waves on highways [4].

While there are different approaches to training robust DRL policies in other domains with different levels of success, our approach is to systematically search for a robust policy by varying the training conditions, evaluating the learned policy on our proposed test set in a single-lane merge scenario, and selecting the highest performing one. The highest performing policy outperforms the human-only baseline with as few as 1 % AVs across different traffic conditions in the single-lane merge scenario.

We further investigate the policy's generalization to more complex scenarios it has not seen during training, specifically a scenario with two merging ramps at a variety of distances, and a merge scenario with a double-lane main road, with cars able to change lanes. Notwithstanding negative prior results showing that a policy developed in a single-lane ring road fails to mitigate the congestion on a double-lane ring road [5], our learned policy outperforms human-only traffic and effectively mitigates congestion in these more complex scenarios as well.

Inspired by our successful results in a high-fidelity microsimulation, this article further contributes a novel extension of the well-known Cell Transmission Model (CTM) that, unlike past CTMs, is suitable for modeling congestion in traffic networks, and is thus suitable for studying congestion-reduction policies such as those considered in this article. Taken together, this article's contributions and insights take us a step closer towards making the exciting concept of traffic congestion reduction through AV control a practical reality.

The rest of the article is structured as follows. Section 2 presents related work. Section 3 provides a background that includes a formalization of the traffic reduction problem, a description of the DRL setup, and a description of our robustness evaluation conditions. Section 4 describes how the DRL policy is learned and analyzes its empirical performance. Section 5 describes the generalization of our policy to unseen, complex roads. Section 6 introduces a novel Cell Transmission Model formulation and use it to empirically characterize the operation of congestion reducing policies. Section 7 presents the hyper-parameters used by the training algorithm and the Cell Transmission Model. The code that generates all data used in this study is available at https://github.com/yulinzhang/MITC-LARG.

2 Related work

Traffic optimization has long been a challenging research area with direct real-world impact [6]. An important research question is how to mitigate highway *stop-and-go waves*, which have been demonstrated to emerge when vehicle density exceeds a critical value, and to result in reduced throughput and increased driving time [2]. In small-scale field experiments, vehicles controlled autonomously by hand-designed driving policies successfully dissipated stop-and-go waves, thus reducing congestion [3]. The industry-wide development of autonomous vehicles (AVs) has inspired researchers to tackle this problem at a larger scale.

Recent progress in Reinforcement Learning (RL) [7] has made it possible to learn congestion reduction AV driving policies that perform well in simulation. Using state-of-the-art algorithms, significant congestion reduction was achieved both in circular roads with a fixed set of vehicles (referred to as *closed* road networks), and acyclic roads with vehicles entering and leaving the system (referred to as *open* road networks) [8–10], as compared with simulated human-driven traffic implemented with accepted human driving models [11]. Most of these past successful driving policies controlled AVs in a *centralized* manner, where a single controller simultaneously processes all available sensing information and sends driving commands to the AVs. More recent efforts focused on developing *decentralized* driving policies

which might be harder to learn, but are considered a more realistic option for real-world deployment, as they mostly rely on local sensing and actuation capabilities [10, 12]. this article continues the line of research on decentralized policies but aims to develop one that is robust to real-world traffic conditions of practical interest.

Recent RL techniques for developing robust policies include adversarial training [13] and domain randomization [14]. Existing research uses these ideas to build congestion reduction policies that are robust to some particular traffic conditions. Wu et al. present policies that can generalize on a closed ring road to traffic densities higher and lower than the ones they were trained on, by randomizing densities during training [15]. Parvate et al. evaluate the robustness of a *hand-coded* controller over different AV penetration and driving aggressiveness [16]. This article focuses on learning a driving policy that is robust to different traffic flows, AV penetrations, AV placement within traffic, and road geometries.

In contemporary unpublished work [17], Vinitsky et al. studied a similar setup. In particular, similarly to our work, they developed a robust, decentralized policy that is shared among all AVs for an open road network scenario. On the other hand, our work differs from theirs in several ways. First we focus on merge scenarios, while they focus on bottleneck scenarios. Second, they developed a robust policy by randomizing the training conditions, while we did a systematic sweep of the training conditions to understand how each training condition contributes to the performance of the trained policy. Third, we further examined the robustness of the policy trained from a merge scenario on a more complex road with multiple merging ramps and multiple lanes.

Finally, to evaluate proposed traffic systems more efficiently, traffic engineers often make use of more abstract traffic models for their initial analyses, such as Cell Transmission Models (CTMs) [18]. Unfortunately, traditional CTMs are not applicable to the topic of this article because they do not capture the traffic congestion from multiple merging inflows. To alleviate this limitation, in Section 6 we introduce a novel CTM formulation that models the traffic congestion by conditionally discounting the merging inflows.

3 Background and setup

We start by introducing the problem of learning a robust traffic congestion reduction policy.

3.1 Road-merge congestion reduction

Consider a network with a main highway and a merging road, as shown in Figure 1c. There are vehicles joining and leaving the network, and the traffic consists of both human-driven and autonomous vehicles. The human drivers are assumed to be self-interested and optimize their own travel time, while autonomous vehicles (AVs) are assumed to be altruistic and have a common goal of reducing traffic congestion. Our goal

is to come up with a driving policy that controls each AV such that traffic performance is improved.

We measure the performance of policies in terms of both *outflow* and *average speed*. Outflow is the number of vehicles per hour exiting the simulation, representing system-level throughput. The average speed represents the time delay it takes an average driver to drive the simulated road. We note that it is important to report both metrics, since scenarios with low and high average speeds could have the same system throughput, such that one is considered congested while the other is not.

A policy can be hand-programmed or learned. Reinforcement learning (RL) has been shown to produce superior policies [8–10, 19] and is therefore our method of choice. Congestion reduction driving policies can either be *centralized*, controlling all vehicles simultaneously based on global system information, or *decentralized*, controlling each vehicle independently based on its local observations. Decentralized policies with no vehicle-to-vehicle communication are most realistic, since they mostly rely on local sensing and actuation capabilities [12, 17], and are therefore the focus of this article.

This multiagent traffic congestion reduction problem can be modelled as a discrete-time, finite-horizon decentralized partially observable Markov decision process (Dec-POMDP) [20], denoted as a tuple $(\mathcal{S}, \{\mathcal{A}_i\}, P, R, \{\Omega_i\}, \mathcal{O}, T, \gamma)$ where,

- $\mathcal{S}$ is a state space representing the location and speed of every vehicle in the network,
- $\{\mathcal{A}_i\}$ is a joint action space for all agents, where $\mathcal{A}_i \in \mathbb{R}$ is a real number that specifies an acceleration action for agent i,
- $P : \mathcal{S} \times \{\mathcal{A}_i\} \times \mathcal{S} \to [0, 1]$ is a stochastic state transition function, which specifies the probability distribution of target state given the source state and action taken by the vehicle. In this paper, this state transition function is realized via a traffic simulator.
- $R : \mathcal{S} \times \{\mathcal{A}_i\} \to \mathbb{R}$ is a global reward function,
- $\{\Omega_i\}$ is a collection of local observations for each agent (see Section 3.2),
- $\mathcal{O} : \mathcal{S} \times \{\mathcal{A}_i\} \times \{\Omega_i\} \to [0, 1]$ outputs the probability that each agent receives a specific observation given the next state and the joint action just taken,
- T is the episode length,
- $\gamma \in [0, 1]$ is the discount factor of reward.

A decentralized, shared *driving policy* is a probability density function over the action space $\pi_\theta : \{\Omega_i\} \times \{\mathcal{A}_i\} \to [0, 1]$ parameterized by θ that stochastically maps each agent's local observations to its driving actions.

Throughout this article we use the SUMO traffic simulator [21] as the state transition function. SUMO is a high-fidelity micro simulator that includes accepted human driving models [11, 22] with configurable traffic networks, flows, and driving aggressiveness, as well as mechanisms for enforcing traffic rules, safety rules, and basic physical constraints. To learn AV driving policies, we use the RLlib library [23]. We

interface with SUMO and RLlib using UC Berkeley's Flow software [24].

3.2 RL-based decentralized driving policy

To learn a decentralized driving policy we use the Proximal Policy Optimization (PPO) algorithm [25]. To facilitate data and computational efficiency and reduce the risk of overfitting, all AVs learn and execute a single, shared driving policy. The observation space and reward design used in this article are modeled after those used by Cui et al. [12], which were shown to be effective for decentralized policies. The observation for each AV includes

- the speed and distance of the closest vehicles in front of and behind it,
- the AV's speed,
- the AV's distance to the next merging point,
- the speed of the next merging vehicle and its distance to the merge junction (assumed to be obtained by the vehicle's cameras/radars, or be computed by some global infrastructure and then shared with all the vehicles).

The reward of the ith AV at time step t is defined as:

$$r_{i,t} = (1 - \mathbb{I}\{done\}) \left(-\eta + (1-\eta) \times \frac{\sum_{j=1}^{n_t} v_j}{n_t V_{max}} \right) + \mathbb{I}\{done\} \cdot Bonus$$

where $\mathbb{I}\{done\}$ is an indicator function of whether an AV is leaving the network; $Bonus$ is a constant reward for an AV when it exits the network; the term $\frac{\sum_{j=1}^{n_t} v_j}{n_t V_{max}}$ represents the normalized average speed, where v_j is the speed of vehicle j, n_t is the total number of vehicles in the network at time t, $V_{\max}$ is the max possible speed, and η is a constant that weights the individual and the global reward.

3.3 Robustness evaluation conditions

Similarly to past work, our baseline setup consists of simulated human-driven vehicles only, where the AVP is 0. In contrast to past work, which typically showed improvement over this baseline in a *single* combination of traffic conditions, our goal is to develop a robust AV driving policy that improves over this baseline across a *range* of realistic traffic conditions, characterized by:

- *Main Inflow Rate*: the amount of incoming traffic on the main artery (veh/hour),
- *Merge Inflow Rate*: the amount of incoming traffic on the merge road (veh/hour),
- *AV Placement*: the place where the AVs appear in the traffic flow; the AVs can either be distributed evenly or randomly among the simulated human-driven vehicles.
- *AV Penetration*: the percentage of vehicles that are controlled autonomously,
- *Merge road geometry:* the distance between two merge junctions (in relevant scenarios), and the number of lanes.

In this article, we focus on a merge inflow rate of 200 veh/hour and a main inflow rate in the range of [1600, 2000] veh/hour since these values tend to

lead to congestion in the baseline (AVP=0) conditions. We vary all the other parameters as follows: AV penetration (AVP) is set to be within [0, 40] percent to represent a realistic amount of controllable AVs that can be expected in the coming years, and the placement of the AVs can either be random or even. For *even placement*, AVs are placed every N human-driven vehicles in a lane. For *random placement*, AVs are placed randomly among simulated human-driven vehicles. Merge road geometries include one or two merges at distances that vary between [200, 800] meters, and the main road can have one or two lanes.

4 Learning a robust policy in the single-lane merge scenario

While real-world congestion-reducing driving policies need to operate effectively in a wide variety of traffic conditions, most past research has tested learned policies under the same conditions on which they were trained. Since in the real world it is impractical to deploy a separate policy for each combination of conditions, our primary goal is to understand whether it is feasible to learn a *single* driving policy that is robust to real-world variations in traffic conditions.

The performance of an RL-based driving policy depends on the traffic conditions under which it is trained. We hypothesize that the policy trained under high inflow, medium AV penetration, and random vehicle placement is robust in a range of traffic conditions defined in Section 3.3 for a single-lane merge scenario. We test this hypothesis by comparing 30 policies, each of which is trained under a combination of traffic conditions specified below in Section 4.1. The training of each policy takes about 7 hours on a $3.7\,\mathrm{GHz}$ Intel 12 Core i7 processor. SUMO has built-in stochasticity which includes vehicle departure times and vehicle driving dynamics. Hence, each policy, including human-only baseline, is evaluated 100 times using a fixed set of 100 random seeds, and each evaluation takes about one hour. After we identify a policy that generalizes well across traffic conditions in the training road geometry, a later section will describe an evaluation this policy on more complex road geometries unseen at training time.

4.1 Discretization of traffic conditions for training

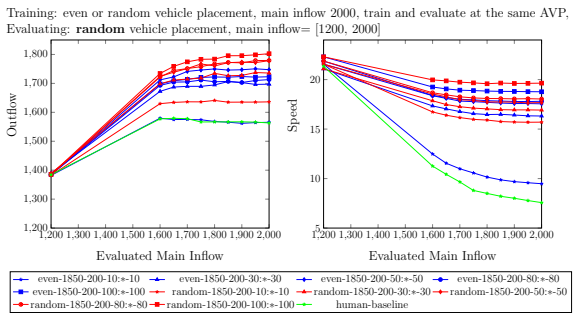
Since there is an innumerable set of possible traffic conditions, for the purpose of training we discretize traffic conditions along their defining dimensions to a total of 30 representative combinations of conditions, as follows. We consider main inflows of 1650, 1850, and 2000 $\mathrm{veh/hour}$ which result in low, medium, and high congestion. We discretize AV placement in traffic to be random or even-spaced. Finally, we discretize the training AV penetration into 5 levels: $10\,\%$, $30\,\%$, $50\,\%$, $80\,\%$, $100\,\%$. Based on this $3 \times 2 \times 5$ discretization, we train 30 policies, one for each combination.

Each trained policy is then evaluated across the range of traffic conditions described in Section 3.3, leading to two performance values (outflow and average speed) on each testing condition for each policy. We plot these results using the following convention. The label of a data point consists of two parts: (i) the training conditions of the policy to be evaluated, and (ii) the policy's evaluation conditions. The policy's training conditions indicate the vehicle placement, main inflow, merge inflow, and AVP, separated by "-". For example, "random-2000-200-30" denotes the policy trained under random vehicle placement with main inflow $2000\,\mathrm{veh/hour}$, merging inflow $200\,\mathrm{veh/hour}$, and $30\,\%$ AVP. The evaluation conditions also consist of vehicle placement, main inflow, merging inflow, and AVP. In this article, the merging inflow is always fixed to be $200\,\mathrm{veh/hour}$ and the vehicle placement is specified separately from the graph label. Therefore we only specify the evaluation-time main inflow and AVP to indicate the evaluation condition for each data point. Hence, each evaluation result is labeled as a 6-tuple, where the first four elements describe the training conditions and the remaining two describe the evaluation conditions. For example, "random-2000-200-30:1800-10" labels the result of policy "random-2000-200-30" evaluated under main inflow $1800\,\mathrm{veh/hour}$ and AVP $10\,\%$. We further use "$*$" in the evaluation condition to denote which evaluation condition varies in a plot. For example, "random-2000-200-30:1800-$*$" indicates that the policy "random-2000-200-30" was evaluated under main inflow of 1800 and varying AVPs; "random-2000-200-30:$*$-10" indicates that policy "random-2000-200-30" was evaluated under AVP $10\,\%$ and varying main inflows.

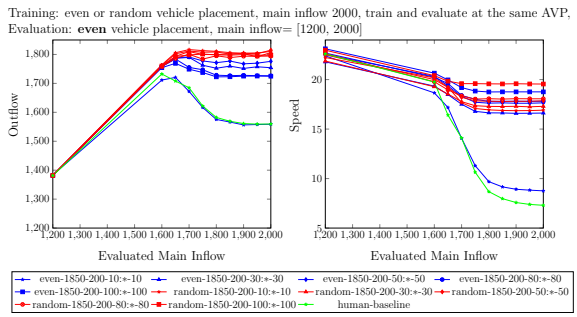
4.2 Robustness to vehicle placement, AV penetration and inflow

In this section, we test our hypothesis that training with high inflow, medium AV penetration, and random vehicle placement yields a robust policy, by showing representative slices of the evaluation results.

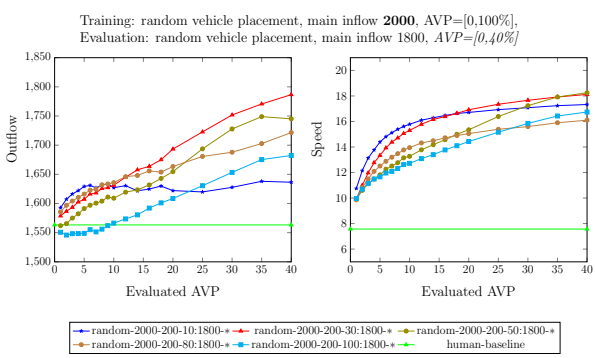
We start by showing that the policies trained under random vehicle placement outperform policies that are trained under even vehicle placement. The performance of a representative subset of these policies is depicted in Figure 2a and 2b. The red curves represent the evaluation results for the policies trained under random vehicle placement, and the blue curves represent the results for the policies trained under even vehicle placement. These policies are evaluated using the outflow and average speed metrics under both random vehicle placement (Figure 2a) and even vehicle placement (Figure 2b). When evaluating on either random placement or even placement, the policies trained with random placement outperform the human baseline as well as their counterparts trained with even placement. Specifically, the results in Figure 2a confirm the intuition that when evaluated with random vehicle placement, the policies trained under random vehicle placement should have better



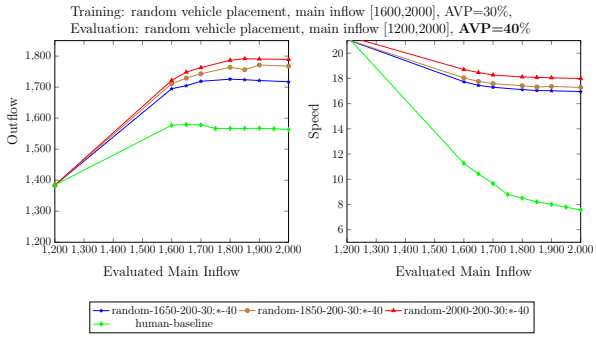
(a) Evaluating the policies with *random* vehicle placement: the policies trained under random placement (colored as red) outperform the policies trained under even placement (colored as blue).



(b) Evaluating the policies with *even* vehicle placement: the policies trained under random placement (colored as red) outperform the policies trained under even placement (colored as blue).



(c) Evaluating the policies trained across different AVPs: the policies trained under AVP 30 % (colored as red) outperform others trained under different AVPs.



(d) Evaluating the policies trained across different main inflows: the policies trained under main inflow 2000 veh/hour (colored as red) outperform others trained under different inflows.

Fig. 2: Results of policies trained under different AV placements, AV penetrations, and main inflows. Figure (a)–(b): we show that the policies trained under random vehicle placement outperform their counterparts trained with even placement, when evaluated under both random and even vehicle placement. Figure (c): we fix the evaluation inflow at a medium level and find that a training AVP of 30 % is the most robust when varying evaluation AVPs; Figure (d): we fix the evaluation AVP, and verify that main inflow 2000 veh/hour is the most robust when varying evaluation inflows.

performance than their counterparts trained with even vehicle placement. However, counter-intuitively, random placement at training time also results in more robust policies when testing under *even* placement. We hypothesize that this performance increase is due to the more diverse data collected when RL vehicles are randomly placed.

Next, we confirm the intuition that the polices trained under medium AV penetration are better than others. Figure 2c show when fixing the main inflow, the policies trained under AVP 30 % (red curve with triangle) are competitive in both their outflow and average speed when evaluated under varying AVPs. They have the best performance across a large range of

the evaluation AVPs. We hypothesize that these mid-range AVP values during training perform best since (i) lower AVP may not encounter enough situations with densely distributed AVs, and (ii) higher AVP may not encounter enough situations with sparsely distributed AVs.

Finally, we test the hypothesis that the policies trained under high inflow are robust. When fixing the AVP and varying main inflow during evaluation, Figure 2d shows that the policy trained under main inflow $2000\,\mathrm{veh/hour}$ (red curve) has better performance than policies trained with different main inflows, in terms of both outflow and average speed. We hypothesize that the policies trained under the highest inflow outperform others because a higher main inflow yields more diverse vehicle densities at training time. Specifically, the simulation dynamics can lead high inflow to include both dense and sparse vehicle placement, while a lower main inflow tends to mostly result in a sparse vehicle distribution.

Verifying our hypothesis, we find that the policy "random-2000-200-30", which is trained under random vehicle placement, main inflow $2000\,\mathrm{veh/hour}$, merge inflow $200\,\mathrm{veh/hour}$, and AVP $30\,\%$, outperforms the alternatives in terms of robustness. In the single-lane merge scenario, this policy achieves significant improvement over the human-only baseline across all evaluating conditions when the AVP is greater than or equal to $1\,\%$ during deployment (with p-value 0.05 as the cutoff for significance).

5 Deploying the learned policy to more complex roads

We learned a robust policy in a single-lane merge scenario. To push this policy one step further toward a real-world deployment, we test this policy's robustness to more complex road structures: roads with two merging ramps, and double-lane roads.

5.1 Deployed to roads with two merging ramps

We first deploy the selected policy on more complex road structures, which have two merging roads at varying distances as shown in Figure 3, and evaluate the performance of the learned policy with respect to the distance between these two ramps.

Consider the merge scenario with two merging ramps: the first merging ramp is located 500 meters from the simulated main road's start, the second merging ramp is located 200, 400, 600, or 800 meters after the first, the total length of the main road is 1500 meters, and the total length of the merging roads is 250 meters. We tested the random-2000-200-30 policy with random AV placement, main inflow of $1800\,\mathrm{veh/hour}$, merge inflow $200\,\mathrm{veh/hour}$, across a range of AV penetrations and the above gaps between the two merging roads.

The results are shown in Figure 4, where the blue curves show the performance of the policy to be tested with different AVP values, and the red curve shows

Fig. 3: A merge road with two merging on-ramps.

the human baseline's performance. The random-2000-200-30 policy is best when the distance between the two on-ramps is large. As we decrease this distance, the performance gap from the human baseline decreases, but remains positive even when the merging ramps are just 200 meters apart, which is the setup that is most different than the training conditions, as explained next. When the distance between on-ramps is small, the traffic congestion at the second merging ramp interferes with the traffic flow at the first merging ramp, but is not observable to the RL vehicles approaching the first ramp. As we increase the distance between these two merging ramps, such interference decreases and the traffic flow approaching these two merging ramps can be treated by the AVs increasingly independently. As a consequence, when these two merging ramps become further away from each other, the decision making processes for the AVs become similar to those on the single-lane merge roads — they only need to consider the traffic flow at the next incoming junction. To summarize, the selected policy slightly reduces traffic congestion in the two-ramp scenario; and its performance improves as the distance between these two ramps increases.

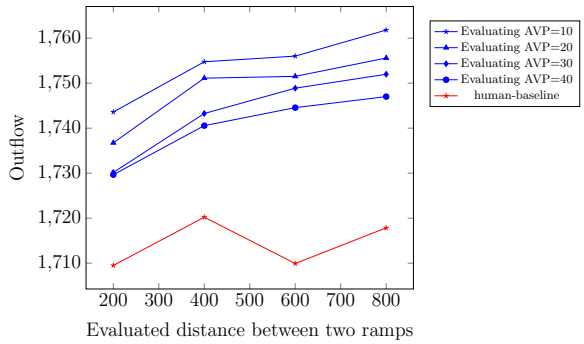


Fig. 4: Results of deploying the selected training policy on roads with two on-ramps. The result of human-only traffic is represented as red, and the results of the learned policy are represented as blue.

5.2 Deployed to double-lane merge roads

Urban highways often consist of multiple lanes. Thus past research suggesting that AVs might *increase* traffic congestion on multi-lane roads [5] has (rightfully) raised concerns about the practical deployability of systems like the one considered in this article. Contrary to those results, we find that AVs can reduce congestion even in multi-lane scenarios. Specifically, we consider a double-lane merge road, by adding a second lane in the main road, as shown in Figure 5. Similarly to the single-lane merge scenario, the vehi-

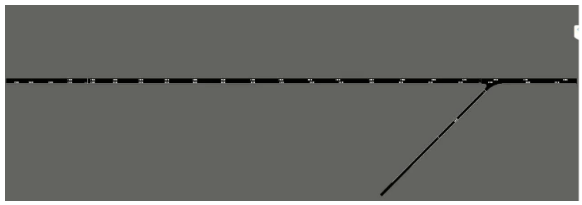


Fig. 5: A double-lane merge scenario.

cles in the right lane must yield to the vehicles from the merging lane and may cause potential congestion in the right lane. But the vehicles in the left lane have the right of way when passing the junction. As a consequence, the vehicles in the left lane tend to move at a faster speed, and there will be more vehicles changing from right to left for speed gain than the number of vehicles changing from left to right. Those lane-changing vehicles cause additional stop-and-go waves in the left lane.

We test the robustness of our selected policy when deployed in the right lane in this new road structure. In our experiments, the left lane contains no AVs and an inflow of $1600\,\mathrm{veh/hour}$ human-driven vehicles, and the right lane contains an AVP of $10\,\%$–$40\,\%$ that are controlled by our selected policy. Figure 6 shows that for right main inflows of $1600-2000\,\mathrm{veh/hour}$, our policy improves outflow by about $4\,\%$ and traffic speed by about 2x compared with human-only traffic. We observed that the learned policy, mitigating the congestion in the right lane b also reduces the amount of lane-changing vehicles since the right lane is less congested. Hence, the policy trained on the single-lane merge road generalizes well in the double-lane merge scenario.

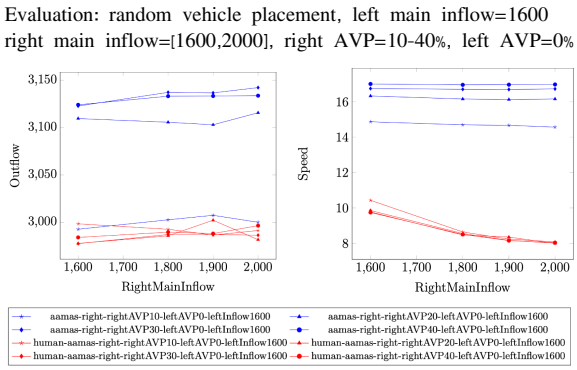


Fig. 6: Results of deploying the selected training policy on the double-lane merge roads. The human-only traffic is represented as red curves, and the traffic controlled by the learned policy is represented as blue curves.

6 Abstract Analysis in an Extended Cell Transmission Model

The findings presented in Sections 4 and 5 mark a significant advancement as they showcase, for the first time, a driving policy that exhibits generalization capabilities across diverse traffic conditions and real-world road structures. This achievement represents a notable stride towards the practical realization of traffic congestion reduction through autonomous vehicle (AV) control. Nonetheless, a knowledge gap persists regarding the extent to which a local driving policy, operating in a distributed manner with independent control for each AV, contributes to overall enhancements in average speed and outflow. Moreover, assessing this driving policy's effectiveness using a high-fidelity microsimulation tool like SUMO poses computational challenges, even on high-performance computing platforms.

As summarized in Section 2, traffic engineers commonly rely on abstract traffic simulators, which efficiently calculate macroscopic traffic behavior without simulating each individual vehicle, to prototype and assess new traffic protocols. Cell Transmission Models (CTM) [18] are widely utilized in such

abstract traffic simulations. However, existing CTMs do not incorporate the modeling of traffic congestion resulting from multiple merging inflows, rendering them unsuitable for our specific research focus. In this section, we present a novel CTM that effectively captures the traffic congestion caused by merging inflows. We validate this model by comparing it to microsimulation outcomes obtained from SUMO. Additionally, we employ this CTM to characterize the operation of our proposed congestion-reducing policies and gain insights about how a local driving policy improves traffic performance globally.

Our analysis proceeds according to the following steps:

- Discretizing the road into basic segments (referred to hereby as *cells*)
- Empirically fitting a fundamental diagram of traffic flow for each cell.
- Using these fundamental diagrams to construct a novel extension of a CTM for the merge scenario in Figure 1.
- Validating this CTM against SUMO by showing that their global behaviors (overall simulation inflow and outflow) are similar.
- Further introducing a novel extension of CTM to model the double-lane merge scenario from Figure 5, and similarly validating its global behavior against SUMO's.
- Using these CTMs to extract insights regarding the desired local (intra-cell/segment) behavior of a policy to improve global traffic flow (simulation outflow), which in turn provides a direction for designing congestion-reduction policies for large-scale multilane scenarios that are too slow to explore by exhaustive simulations.

6.1 Discretizing road into cells and fitting their fundamental diagrams

We start by discretizing the single-lane merge scenario from Figure 1 into $100\,\mathrm{m}$ cells, as shown in Figure 7. The cell length of $100\,\mathrm{m}$ was selected to be small enough to capture the local traffic around each autonomous vehicle, and large enough for computational efficiency.

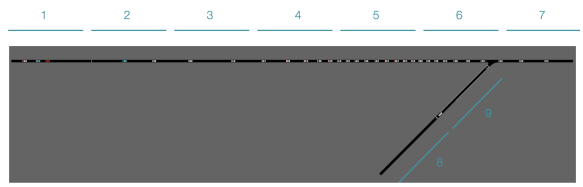
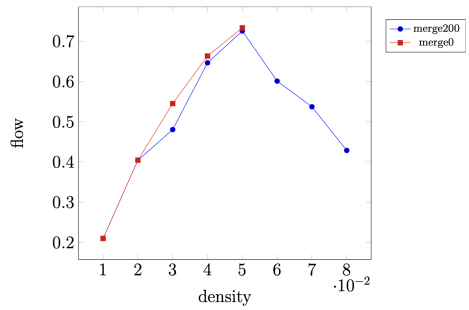


Fig. 7: Discretizing the road into cells.

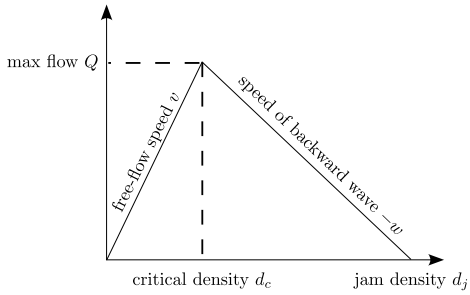
Next, we import from traffic flow theory the concept of a traffic fundamental diagrams, which yields the relationship between the traffic density and traffic flow [26]. To obtain a fundamental diagram for each cell in SUMO, we profiled the instantaneous density and average speed, and calculated the flow as the product of instantaneous density and average speed. Since the fundamental diagram characterizes the intrinsic properties of the road conditions (such as capacity and speed limit), the diagram is independent of the

inflows. In Figure 8a, we profile the fundamental diagram of merge inflow 200 (blue) and 0 (red) veh/hour. For merge inflow 0, there is no congestion in the road and so the density of the cells will never be higher than 0.05 veh/m. From this fundamental diagram, we can observe that the results for both of these merge inflows are almost the same. Similarly, we observe the same fundamental diagrams for all cells, and therefore we model every cell with the same fundamental diagram.

Based on the observed data, we see that the fundamental diagram is close to a triangular shape. Hence, we fit a triangular fundamental diagram as shown in Figure 8b, which is defined by the slope before the peak (called free-flow speed v), maximum flow Q and its corresponding density (critical density d_c), slope after the peak (speed of the backward wave w), and the density to reach 0 flow (jam density d_j).



(a) The flow density relation under different merge inflows (red curve represents the result of 0 merge inflow, and blue curve represents the result of merge inflow 200 $\mathrm{veh/hour}$.) The horizontal axis is the density ($\mathrm{veh/m}$), and the vertical axis represents the flow ($\mathrm{veh/s}$).



(b) A triangular fundamental diagram and its parameters.

Fig. 8: Profiling the flow density relation of a cell in SUMO, and modelling it as a triangular fundamental diagram.

6.2 Constructing an extended CTM from fundamental diagrams.

Next, we introduce an extended CTM, which models a single-lane merge scenario using the fitted fundamental diagram as a model of intra-cell behaviors. We start by defining two additional parameters that characterize all cells:

- $Q = d_c \times v$ is the maximum number of vehicles that can flow into a cell when the clock advances,
- $N = 100 \times d_j$ is the maximum number of vehicles in a cell, where 100 is the cell length.

Let $y_i(t)$ and $n_i(t)$ be the inflow and number of vehicles in cell i at time t. The inflow is upper bounded by the total number of vehicles in the upstream cells, maximum number of vehicles that can flow into the current cell, and the number of available positions in the cell discounted by the ratio of wave and free-flow speeds [27] i.e.,

$$y_i(t) = \min\left\{n_{i-1}(t), Q, \frac{w}{v}[N - n_i(t)]\right\}$$

When the merge traffic exceeds a certain threshold, more vehicles on the main road will have to slow down

or stop to yield to merging traffic. This causes a reduction in the inflow right after the junction, i.e., at cell 7. To model this, we introduce a conditional penalty factor α to discount the inflow of the cell after the merge: if the flow from the merge road is larger than some threshold β, then the inflow of the downstream cell is discounted by α, i.e.,

$$y_7'(t) = \alpha \times y_7(t),$$

where both α and β are hyper-parameters.

Using the above rules, we can update the number of vehicles at cell i at time $t+1$ by adding the inflow and subtracting the outflow at time t:

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t) \qquad (1)$$

The scenario's overall inflow and outflow are then the inflow of the left most cell (cell 1) and outflow of the right most cell (cell 7). The video of the CTM simulation for single-lane merge scenario can be found here: https://tinyurl.com/single-lane-ctm.

6.3 Validating the single-lane CTM against SUMO

To validate our novel single-lane merge CTM, we run a CTM simulation by iterating the operation suggested by Equation (1) until the inflow and outflow converge to their steady state, and then compare its overall inflow and outflow with SUMO's. Figure 9 shows this comparison, where each data point for SUMO is collected by running 100 simulations, each with a different random seed, and each data point for CTM is collected from a single simulation (since CTM is deterministic). The CTM outflows mostly fall within the 95% confidence bounds of the mean, which represent 100 vehicles or fewer (around 5-6% of the flow), thus providing reasonable similarity between the inflow-outflow plots of the CTM and SUMO. Both curves have similar values as the outflow first increase with inflow, then decreases as the traffic congestion develops, and finally saturates as we further increase the inflow.

Running a CTM simulation takes less than a second, while running 100 SUMO simulations can take minutes, or even hours or days for large scenarios. Therefore, CTM based on the triangular fundamental diagram can be viewed as a lower-fidelity but more computationally efficient alternative for SUMO.

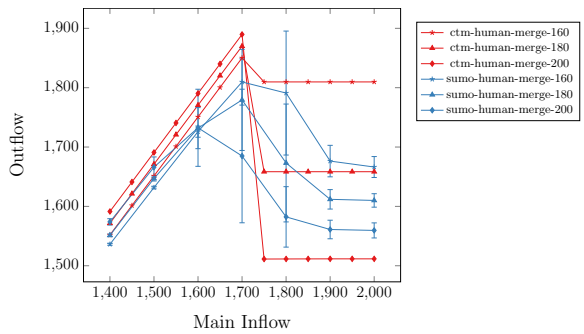


Fig. 9: Comparing the inflow-outflow relation between SUMO and CTM under different main inflows and merge inflows. The range of the main inflow is [1400, 2000], and the range of the merge inflow is [160, 200]. The human-only result in SUMO is represented as cyan curves, and that of CTM is represent as red curves.

6.4 Extending CTM to a double-lane merge scenario

Next, we introduce another novel extension of CTM, modelling for the first time a multilane merge scenario. First, we discretize the double-lane scenario from Figure 5 into $100\,\mathrm{m}$ cells, as illustrated in Figure 10. Next, to capture traffic changing from neighboring cells, we add the following definitions:

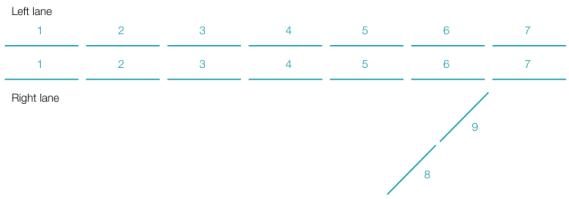


Fig. 10: Discretizing the double-lane scenario from Figure 5 into CTM cells.

- $n_i^l(t), n_i^r(t)$: the number of vehicles on the left and right lanes of cell i at time t
- $lc_i^l(t), lc_i^r(t)$: the number of lane-changes to the left and right lanes of cell i at time t

We then add the following rules:

- The right main road follows the same update rules as that of single-lane case.
- The left main road will not be blocked by the merging vehicles.
- Rules for lane-changing vehicles $lc_i^l(t)$ and $lc_i^r(t)$ from current lane to the target lane:
  - If the number of vehicles in the current lane is less than or equal to that of the target lane, then more vehicles will be motivated to stay and the number of vehicles changing from current lane is small and denoted as ϵ.
  - If the number of vehicles in the current lane is larger than that of the target lane, then additional vehicles will be motivated to change to the less congested lane. Here, we introduce a lane change factor δ, to capture the fraction of vehicles that are motivated to change lanes:

$$lc_i^l(t) = \delta \times (n_i^r(t) - n_i^l(t)) + \epsilon$$

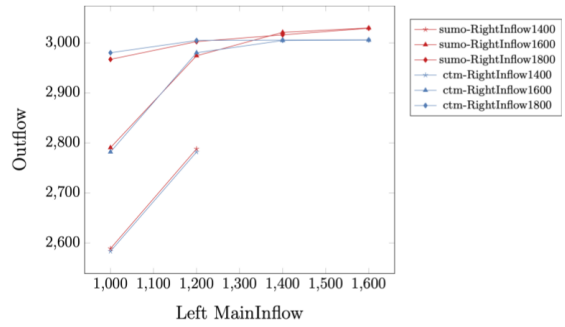
- To capture the traffic congestion caused by lane-changing behaviors, we build flow discounting rules similar to those of the single-lane case as follows. If the number of vehicles changing to cell i ($lc_i^l(t)$) is larger than 0 and the existing number of vehicles ($n_i^l(t)$) is larger than a certain threshold, then there will be congestion caused by lane changing and we discount the outflow using the previously introduced discounting factor α:

$$y_{i+1}(t) := \alpha \times y_{i+1}(t)$$

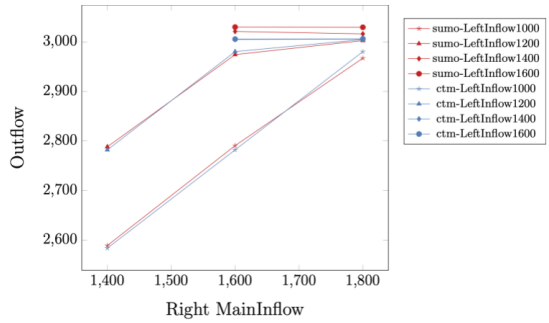
Based on the rules above, we can obtain a double-lane CTM, and a video of this model can be found: https://tinyurl.com/double-lane-ctm.

Similarly to the single-lane CTM, we validate the CTM by iterating the update equation until convergence of inflow and outflow, and then compare its

overall inflow and outflow with SUMO's. Figure 11 shows this comparison on a range of inflows and outflows, where the main inflow on the right lane is chosen to be larger than that of the left lane so that most traffic changes from the right lane to the left lane to reflect a typical merge scenario. It can be seen that the inflow-outflow curves match each other well. We conclude that the double-lane CTM that uses a triangular fundamental diagram to model each cell can serve as a lower-fidelity, computationally efficient alternative to SUMO for the double-lane merge scenario.



(a) Fixing a few right-lane inflows, varying left-lane inflows



(b) Fixing a few left-lane inflows, varying right-lane inflows

Fig. 11: Comparing the inflows and outflows of the double-lane CTM with SUMO's. The results in SUMO are represented as red curves, and the results in CTM are represented as blue.Here we only present the data points where the inflow on the left lane is smaller than that from the right lane.

6.5 Insights from fundamental diagrams and CTM

We introduced novel CTMs for single-lane and double-lane merge scenarios, by discretizing these roads into cells that are simulated using fitted triangular fundamental traffic flow diagrams. We have observed that the inflow-outflow CTM plots approximate closely those of the SUMO micro-simulation, in both single-lane and double-lane merge scenarios. So the CTMs can be treated as a low-fidelity alternative of the SUMO microsimulator. In this section, we present insights about congestion reduction policies that are suggested by studying the behavior of our extended CTMs.

In the triangular fundamental diagram shown in Figure 8b, the flow of each cell is maximized when the density is around a *critical* density in which a maximal flow is achieved (the peak of the rectangle in Figure 8b). Hence, it seems that an effective AV driving policy ought to seek to manipulate the traffic density in its vicinity to remain close to the critical density. Indeed, our proposed driving policy does so by slowing down to reduce traffic density if there is congestion ahead.

A similar intuition applies in the double-lane merge scenario as well. According to the lane-changing rules of CTM and SUMO, vehicles change from high-density lanes to low-density lanes.

Autonomous vehicles are observed to encourage such lane-changing behaviors, by opening a gaps suitable for other cars to merge into. This behavior helps to optimize the traffic density in both lanes toward their critical densities.

The benefit of our extended CTMs could become even more apparent in large-scale multilane scenarios that are too slow to explore by exhaustive simulations of different traffic conditions. Using a similar approach, we can discretize such scenarios into cells modelled using fitted fundamental diagrams, and then use the computationally-efficiently CTMs to explore a range of traffic conditions and desired AV density control policies, which could direct the development of practical congestion reduction policies for large-scale scenarios.

7 Implementation Details and Hyper-parameters

All experiments are built on top of SUMO 1.6.0 and UC Berkeley's Flow software framework [24]. The human-driven vehicles are controlled by the Krauss model with hyper-parameters defined in Table 1. To control the autonomous vehicles, we use Proximal Policy Optimization algorithm [25] to learn a driving policy, and the hyper-parameters for this algorithm is defined in Table 2. The hyper-parameters used by CTM is shown in Table 3. Our implementation is available at https://github.com/yulinzhang/MITC-LARG.

8 Conclusion and future work

We presented an approach for learning a congestion reduction driving policy that performs robustly in road merge scenarios over a variety of traffic conditions of practical interest. Specifically, the resulting policy reduces congestion in AV penetrations of $1\,\%$–$40\,\%$, traffic inflows ranging from no congestion to heavy congestion, random AV placement in traffic, single-lane single-merge road, single-lane road with two merges at varying distances, and double-lane single-merge road with lane changes. The process of finding this policy involved identifying a single combination of training conditions that yields a robust policy across different evaluating conditions in a single-lane merge scenario. We find, for the first time, that the resulting policy generalizes beyond the training conditions and road geometry it was trained on.

Recently there has been an increasing interest in developing RL training methods that result in robust policies. In our domain we find that randomizing AV placement and searching for an effective training setup over the space of traffic conditions achieve robustness effectively. The straightforward nature of our method and its limited set of assumptions and tuning parameters make it a potential candidate for real-world deployments. Given that RL algorithms have been shown to be brittle in many domains, finding an RL-based policy that performs robustly across

Table 1: Hyper-Parameters for Human-driven Vehicles

Parameter	Value
Controller	IDM Controller
Max Acceleration	2.6
Max Deceleration	4.5
Expected Time Headway	1 second

Table 2: Hyper-Parameters for Training Autonomous Vehicles

Parameter	Value
Algorithm	Proximal Policy Optimization (PPO)
Horizon	14000
Simulation Time Step Size	0.5
Optimizer	Stochastic Gradient Descent
Learning Rate	piece-wise linearly decreasing starting from 5×10^{-4} (From scratch)
Discount Factor (γ)	0.998
GAE Lambda (λ)	0.95
Actor Critic	True
Value Function Clip Parameter	10^8
Number of SGD Update per Iteration	10
Model hiddens	[100,50,25]
Clip Parameter	0.2
Entropy Coefficient	10^{-3}
Sgd Minibatch size	4096
Train Batch Size	60000
Value Function Share Layers	True
Value Loss Coefficient	0.5
KL Coefficient	0.01
KL Target	0.01
Max Acceleration	2.6
Max Deceleration	4.5
Training Iterations	500
Number of Rollouts per Iteration	30
Bonus	20
η	0.9

a wide variety of traffic conditions in the challenging domain of multiagent congestion reduction is both encouraging and somewhat surprising.

As a secondary contribution of the article, and in order to more rapidly assess potential directions for reducing congestion at merge points, we introduced a novel variant of the Cell Transmission Model (CTM). To this end, we first fit a fundamental diagram for the micro-simulation results in SUMO. Based on this fundamental diagram, we then construct an extended CTM that accounts for traffic congestion in the merge scenario. This extended CTM can serve as a lower fidelity, but more computationally efficient, alternative to micro-simulation, and can thus be leveraged for

Table 3: Hyper-Parameters for the Extended Cell Transmission Model

Parameter	Value
Q	$4.0\,\mathrm{veh/s}$
N	14
v	$21\,\mathrm{m/s}$
w	$8.40\,\mathrm{m/s}$
d_c	$0.04\,\mathrm{veh/m}$
α	0.65
β	1
δ	0.15
ϵ	0.05

rapid prototyping. Additionally, we reflect on insights from experiments using the extended CTM model that motivate training policies that improve the traffic flow by keeping the traffic density close to the critical density from the fundamental diagram.

Nonetheless, our work has a few limitations that could serve as important directions for future research. First, the question of whether there exists a driving policy that reduces congestion when deployed on the left lane of multilane scenarios still open. Second, our tests used the same aggressiveness level for all simulated human-driven vehicles. Testing with a variety of human behaviors would further increase the simulation results' applicability. Third, there is room to investigate a wider variety of road geometries beyond the ones we investigated. Finally, even after investigating these extensions, there will likely be a sim2real gap to close, due to noisy/limited sensing and actuation delay. These limitations notwithstanding, this article's contributions and insights advance our ongoing effort to reduce traffic congestion via AV control in the real world.

9 Acknowledgement

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by the National Science Foundation (FAIN-2019844), the Office of Naval Research (N00014-18-2243), Army Research Office (W911NF-19-2-0333), DARPA, Bosch, and Good Systems, a research grand challenge at the University of Texas at Austin. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

10 Compliance with Ethical Standards

The prior and current affiliations that are in the conflict of interest include The University and Texas at Austin, General Motors, Texas A&M University and Amazon Robotics. The corresponding author is prepared to collect documentation of compliance with ethical standards and send if requested.

References

[1] Lomax, T., Schrank, D., Eisele, B.: 2021 Urban Mobility Report. https://mobility.tamu.edu/umr/. Accessed: 2021-10-07

[2] Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S.-i., Yukawa, S.: Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam. New Journal of Physics **10**(3), 033001 (2008)

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

[4] Mitarai, N., Nakanishi, H.: Convective instability and structure formation in traffic flow. Journal of the Physical Society of Japan **69**(11), 3752–3761 (2000)

[5] Cummins, L., Sun, Y., Reynolds, M.: Simulating the effectiveness of wave dissipation by follower-stopper autonomous vehicles. Transportation Research Part C: Emerging Technologies **123**, 102954 (2021)

[6] Downs, A.: Stuck in Traffic: Coping with Peak-hour Traffic Congestion. Brookings Institution Press, JSTOR (2000)

[7] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, Cambridge, MA (2018)

[8] Wu, C., Kreidieh, A., Vinitsky, E., Bayen, A.M.: Emergent behaviors in mixed-autonomy traffic. In: Conference on Robot Learning, pp. 398–407 (2017)

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

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

[11] Treiber, M., Kesting, A.: The intelligent driver model with stochasticity-new insights into traffic flow oscillations. Transportation Research Procedia **23**, 174–187 (2017)

[12] Cui, J., Macke, W., Yedidsion, H., Goyal, A., Urieli, D., Stone, P.: Scalable multiagent driving policies for reducing traffic congestion. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, pp. 386–394 (2021)

[13] Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.: Robust adversarial reinforcement learning. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 2817–2826 (2017)

[14] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 23–30 (2017). IEEE

[15] Wu, C., Kreidieh, A.R., Parvate, K., Vinitsky, E., Bayen, A.M.: Flow: A modular learning framework for mixed autonomy traffic. IEEE Transactions on Robotics, 1–17 (2021)

[16] Parvate, K.: On training robust policies for flow smoothing (UCB/EECS-2020-197) (2020)

[17] Vinitsky, E., Lichtle, N., Parvate, K., Bayen, A.: Optimizing mixed autonomy traffic flow with decentralized autonomous vehicles and multi-agent rl. ACM Transactions on Cyber-Physical Systems (2023)

[18] Daganzo, C.F.: The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological **28**(4), 269–287 (1994)

[19] Yan, Z., Kreidieh, A.R., Vinitsky, E., Bayen, A.M., Wu, C.: Unified automatic control of vehicular systems with reinforcement learning. IEEE Transactions on Automation Science and Engineering **20**(2), 789–804 (2023)

[20] Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Mathematics of Operations Research **27**(4), 819–840 (2002)

[21] Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of sumo-simulation of urban mobility. International Journal on Advances in Systems and Measurements **5**(3&4) (2012)

[22] Krauß, S.: Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. Technical Report DLR-FB-98-08, German Center for Air and Space Navigation (1998)

[23] Duan, Y., Chen, X., Houthooft, R., Schulman, J., Abbeel, P.: Benchmarking deep reinforcement learning for continuous control. In: International Conference on Machine Learning, pp. 1329–1338 (2016)

[24] Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., Bayen, A.M.: Flow: Architecture and benchmarking for reinforcement learning in traffic control. arXiv preprint arXiv:1710.05465, 10 (2017)

[25] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

[26] Geroliminis, N., Daganzo, C.F.: Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transportation Research Part B: Methodological **42**(9), 759–770 (2008)

[27] Boyles, S.D., Lownes, N.E., Unnikrishnan, A.: Transportation Network Analysis vol. 1, 0.90 edn. (2022)