Multiagent Driving Policy for Congestion Reduction in a Large Scale Scenario

Jiaxun Cui[1] * **William Macke**[1] **Aastha Goyal**[1]

Harel Yedidsion[1] **Daniel Urieli**[2] **Peter Stone**[1,3]

Abstract

Traffic congestion is a major challenge in modern urban settings. The industry-wide development of autonomous and automated vehicles (AVs) motivates the question of how can AVs contribute to congestion reduction. Past research has shown that in small scale mixed traffic scenarios with both AVs and human-driven vehicles, a small fraction of AVs executing a controlled multiagent driving policy can mitigate congestion. In this paper, we scale up existing approaches and develop new multiagent driving policies for AVs in scenarios with greater complexity. We start by showing that a congestion metric used by past research is manipulable in *open road network* scenarios where vehicles dynamically join and leave the road. We then propose using a different metric that is robust to manipulation and reflects open network traffic efficiency. Next, we propose a modular transfer learning approach and use it to scale up the multiagent driving policy to a realistic simulated scenario that is an order of magnitude larger than past scenarios (hundreds rather than tens of vehicles). Our experimental study shows that the resulting policy improves traffic efficiency over human-driven traffic in a large open network, where existing approaches fail to do so. Another key advantage of our modular transfer learning approach is that it avoids collecting samples from entire network, which saves up to 80% of training and data collection time in our experiments.

1 Introduction

Traffic congestion is one of the leading causes of lost productivity and decreased standard of living in urban settings [3]. Real world transportation systems suffer from inefficiency, partly due to the tendency of self-interested drivers to maximize personal utility over social welfare, and the inherent randomness in human driving [21]. The industry-wide development of autonomous and automated vehicles (AVs) motivates the question of how could AVs contribute to congestion reduction. Since AVs are controlled by predefined policies, unlike their human controlled counterparts they can be made to act selflessly to increase the total social welfare. Through strategic driving, AVs can influence the behavior of the human-driven vehicles in a network, and potentially mitigate congestion. For instance, an AV may smoothly slow down when reaching a merge point in the road to allow for another vehicle to merge, and force the human-controlled vehicles behind it to smoothly slow down as well, thus preventing sharp decelerations and stops that could create and propagate back congestion. Past research has shown that in human-driven traffic, a small fraction of AVs executing a controlled multiagent driving policy can mitigate congestion in simplified simulated and real-world scenarios [12, 18]. These AVs executed a learned or programmed driving policy, which controlled their speed and possible lane-changes to mitigate a form of congestion called *stop-and-go waves*. The

*[1] The University of Texas at Austin, [2] GM, [3] Sony AI

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

research focusing on simulated experiments used Berkeley's Flow framework [20], which combines the SUMO traffic simulator [7] with the RLlib deep reinforcement learning library [4]. The simulated scenarios included cyclic road networks with a fixed set of vehicles, and more realistic non-cyclic road networks with vehicles joining and leaving, referred to as *closed road networks* and *open road networks* respectively. In this paper, we use the Flow framework to scale up existing approaches and develop new multiagent policies for open road network scenarios with increased realism and complexity.

Past simulated open road network scenarios were relatively small – a few hundreds of meters long, with a few tens of vehicles. Their small size allowed for using a centralized multiagent driving policy, which is limited in its ability to scale up since the observation and action spaces increase exponentially with the number of AVs. Moreover, we observed that the metric used by past research to show congestion improvement was manipulable by an RL agent in open road networks.

The contributions of this paper are as follows.

- We outline the drawbacks of the time-average sample-average speed metric(for simplicity, we will use average-speed metric or average speed for this term in further discussion) in Section 4.1, and show empirically in Section 6.1 that in open road networks, since incoming vehicle flows can be moderated by the AVs, the average-speed metric used by past research is manipulable by an RL agent.
- We propose to use instead the *outflow* congestion metric (rate of vehicles exiting the network), highlight its advantages over the average-speed metric in Section 4.1. Specifically,we show that this metric is robust to manipulation and reflects open network traffic efficiency.
- Using transfer reinforcement learning, we scale up a centralized multiagent policy to outperform human-driven traffic on a realistic scenario that is an order of magnitude larger than past scenarios (hundreds instead of tens of vehicles), on a real-world road network downloaded from Open Street Maps [9], specifically road I-696 in Michigan, USA. To avoid an exponential increase in complexity we transfer a learned centralized policy by operating it locally around a key location in the larger network.

The rest of the paper is organized as follows. Section 2 surveys related work. Section 3 defines the problem and the notation used. In Section 4 we compare evaluation metrics and outline our proposed solution methods. Section 5 provides an empirical evaluation, the results of which are discussed in Section 6. Section 7 concludes the paper and suggests future work.

2 Related Work

The recent industry-wide development of autonomous and automated vehicles (AVs) has led to a surge of interest in harnessing AVs to reduce traffic congestion. On the theoretical side, there have been efforts to formalize and analyze the foundations for AVs impacting traffic systems [19]. On the applied side, large-scale traffic simulators have been adopted into a newly developed experimental framework called Flow [20, 18], which we use in this paper. Using Flow, past research showed that Reinforcement Learning (RL) [13] can learn an effective centralized multiagent driving policy, which simultaneously senses and controls all AVs, and improves the average traffic speed over human-driven traffic, implemented with accepted human driving models [16, 17]. However, we show that the average-speed metric is manipulable by an RL agent and might not accurately reflect the networks' traffic efficiency, and propose using an alternative metric.

Since using RL to learn controllers in realistic simulated or real-world setups could be impractically slow, some research looked at using transfer learning [14] to expedite learning, by transferring from a simulated ring to a simulated simple merge scenario [8], and from a simulated to a scaled city [5]. Our transfer learning approach is different in the modular way it reuses state representation, which makes it more scalable. In the ring-to-merge transfer source and target scenario structure and size by assuming a maximum number of AVs in the road network, duplicating the ring state representation by this number, and using 0-padding if the actual number of AVs was smaller. In contrast, in our transfer approach the policy does not directly control more AVs than it was trained for. Instead, it is deployed only in a specific key location in the scenario, and its state representation remains the same even though the scenario has a different geometry and larger number of participants. In addition, the policy transferred from ring to merge did not surpass the performance of a policy trained from scratch, while using our approach the transferred policy did. In the transfer learning from simulation

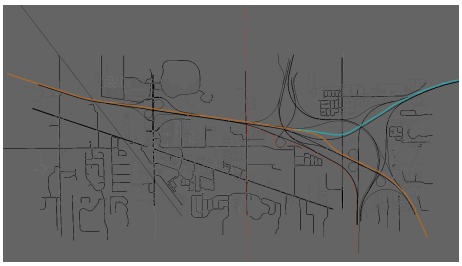

Figure 1: An image of the I-696 highway and intersections, as displayed in SUMO. It is used in our experiments as a representative large, open network.

to a scaled city, the source and target scenarios were the same and the state and action spaces were identical, so the challenge was not to generalize to a different scenario, but instead to compensate for the differences between the simulation and the real world. Using our approach, we were able to scale to a simulated scenario with hundreds of vehicles, an order of magnitude larger than before, specifically road I-696 in Michigan, displayed in Figure 1, infamous for its high congestion.

3 Domain Description and Notation

Problem definition. Given an open road network with mixed autonomy traffic consisting of both human-driven vehicles and AVs, maximize the network's traffic efficiency by controlling the AV accelerations. Traffic Efficiency is measured in terms of **outflow** – the number of vehicles per hour exiting the network. A solution to the congestion reduction problem is a multiagent driving policy which maps the AVs' states to acceleration actions.

Assumptions

- Agents (AVs) are altruistic and have a common goal of reducing system congestion
- Human drivers are self-interested and try to improve their own travel time
- Agents may be able to communicate

MDP Definition The congestion reduction problems we address in this paper can be modeled as a discrete-time, finite-horizon Markov Decision Process (MDP) [10], which is a tuple $M = (\mathcal{S}, \mathcal{A}, P, R, \rho_0, T)$, where $\mathcal{S}$ is a state set, $\mathcal{A}$ an action set, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_+$ a transition probability distribution, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ a reward function, $\rho_0 : \mathcal{S} \to [0, 1]$ an initial state distribution, and T the time horizon. A *driving policy* is a probability distribution $\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$ parameterized by θ that stochastically maps states to driving actions.

In order to find a solution policy, we train an RL agent whose goal is to optimize a driving policy to maximize the expected return $E_\tau \left[\sum_{t=0}^{T} R\left(S_t, A_t\right)\right]$, where $\tau := (S_0, A_0, S_1, A_1 \dots)$ denotes a trajectory, $S_0 \sim \rho_0$, $A_t \sim \pi_\theta\left(\cdot|S_t\right)$, $S_{t+1} \sim P\left(\cdot|S_t, A_t\right)$. In this paper, $\mathcal{S}$ is a set of AV observations, $\mathcal{A}$ a set of acceleration actions, P is computed by the simulator, and R denotes the reward function. We discuss several implementations for the reward function in Section 4.1

Simulation Environment We interface to the SUMO traffic simulator [7] using UC Berkeley's Flow software [6]. Flow provides OpenAI Gym [1] environments as wrappers around SUMO for easy interaction with various RL algorithm implementations. The simulator takes in maps of road structures, and simulates vehicle movements using accepted human driving models [16, 17] and definitions of *inflows*, i.e., the location and rate of vehicles entering the network. The simulated vehicles follow safety and acceleration limits enforced by the simulator. A vehicle's *leader* and *follower* are the closest vehicles in front and behind it (if exist). We note that the actual inflow rate frequently differs from the requested one, for instance in cases where vehicles cannot enter the road network due to congestion. This opens an option for a vehicle to moderate the inflow by slowing down intentionally immediately after joining the network. The inability to guarantee exact inflows is the reason that the average-speed-based metric is not a valid congestion measure in open networks, as discussed in Section 4.1.

4 Methodology

In this section we discuss our training methodology. We begin by describing the evaluation metrics we use, and then describe in detail each of our training techniques.

4.1 Evaluation Metrics

In the Flow benchmark [18], the performance of the system is evaluated using time-average sample-average speed over the episode, defined by Equation 1

$$\text{Time-Average Sample-Average Speed} \triangleq \frac{\sum_{t=1}^{T} \sum_{i=1}^{n_t} v_{i,t}/n_t}{T} \tag{1}$$

where n_t is a time-dependant variable representing the number of vehicles in the traffic network at time t, $v_{i,t}$ is the instantaneous speed of vehicle i at time t, T is the episode length.

In an ideal scenario with constant inflows, there are multiple metrics that would all lead to the same ordering of policies by quality according to a metric: maximizing average speed, maximizing network outflow, and minimizing average time delay [2]. However, average speed and time delay are only equivalent when inflows are identical as they are a function of the number of vehicles observed. In open road networks, vehicles dynamically enter and exit the network. A good policy for open networks should optimize network outflow by maximizing the number of vehicles that pass through the network in a fixed time interval. However, the average-speed metric can be manipulated as it is highly dependent on the inflows of the network. For instance, one way to manipulate the average-speed metric is to block the incoming vehicles from entering the network until there is enough space for incoming vehicles to accelerate to the maximum speed, thus maximizing the average-speed metric by compromising inflows, thus reducing the network outflow. The average-speed metric is vulnerable because of ignoring the resulting slow speeds of vehicles that haven't entered the network. The outflow metric on the other hand is robust to this form of manipulation since it does not have to consider the velocities of the vehicles that have not entered the simulation yet. This reduction of inflows and outflows as a means of improving average speed is demonstrated in Table 1 that compares the results of using three reward functions: the original Flow reward, the average-speed-based, and outflows on Simple Merge defined in Section 6.1. Therefore, we propose **Outflow** as a metric for open networks as defined in Equation 2.

$$\text{Outflow} \triangleq \frac{\sum_{t}^{T} O_t}{T} \tag{2}$$

where T is the episode length and O_t represents the number of vehicles that leave the network during timestep t. Since reducing the number of vehicles on the network will also reduce the number of vehicles that can exit, the Outflows metric is more robust to the problems of inflow manipulation.

4.2 Centralized Solution Approach

We use a centralized approach similar to previous work [18, 8] as a starting point. In the centralized approach, there is a centralized RL agent trained using the Proximal Policy Optimization (PPO) algorithm [11], which controls a predefined fixed number of agents, N_{AV} as illustrated in Figure 2.

PPO is a policy gradient RL algorithm that uses a critic for variance reduction. It learns a stochastic policy that decides which action to take under a particular observed state of the environment. and the policy is optimized using analytic gradient steps that are limited by KL divergence. Both actor and critic are implemented as deep neural nets. RL vehicles are added to the list of controlled vehicles according to a FIFO rule based on when they entered the network. Below we discuss the state space and reward signal used for the centralized approach.

4.2.1 State

The state features of the centralized approach used in [8] and which we also use in our work, include the following features:

1. Normalized speed of the AV_i, v_i
2. Normalized speed of the leader of AV_i, v_i^L

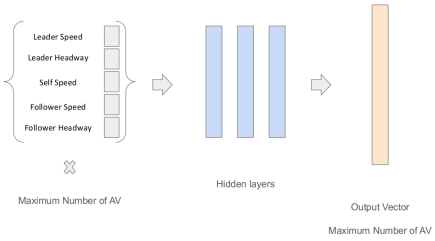

Figure 2: Centralized model, where local states for vehicles are concatenated to form one state. They are passed through a series of hidden layers that result in a final output vector of size N_{AV} which is the number of controlled AVs. Each of the values in this vector is a target speed for the corresponding AV.

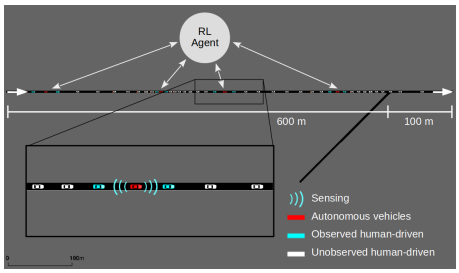

Figure 3: Simple Merge network of length 700 m and inflow rate 2000 veh/hr with an on-ramp of inflow rate 200 veh/h. Perturbations from the on-merge lead to congestion [8]

3. Normalized headway between AV_i and its leader, h_i^L
4. Normalized speed of the follower of AV_i, v_i^F
5. Normalized headway between AV_i and its follower, h_i^F

The speed values are normalized by the max possible speed V_{max}, and the headway values are normalized by a constant representing the maximum possible headway, h_{max}. The state vectors for each controlled vehicle are concatenated into one observation vector, S_t. Suppose the maximum number of AVs controlled by the centralized policy is N_{AV}, then the state feature S_t is a vector of length $5N_{AV}$, and is padded with zeros when the number of AVs in the network is smaller than N_{AV}. Formally, the state of AV_i at time t, $S_{i,t}$ is defined in Equation 3, and the concatenated state of all the AVs at time t, S_t is defined in Equation 4.

$$S_{i,t} = [\frac{v_{i,t}}{V_{max}}, \frac{v_{i,t}^L}{V_{max}}, \frac{h_{i,t}^L}{h_{max}}, \frac{v_{i,t}^F}{V_{max}}, \frac{h_{i,t}^F}{h_{max}}] \tag{3}$$

$$S_t = [S_1, S_2, ...S_{N_{AV}}] \tag{4}$$

4.2.2 Reward

There are several possible objectives to optimize for in open networks, such as maximizing network outflow, or minimizing the maximum time delay of any vehicle to prevent starvation. In this paper, we focus on maximizing the efficiency of a network in the form of average outflows. There are three reward functions considered in our experiments.

Original Flow Reward [18] The reward in the Flow benchmark is composed of ℓ_2-norm distance to a desired velocity and a small-headway penalization term. This reward encourages every vehicle to travel as close as it can to the desired speed every time step while maintaining a large headway.

$$r_t = \max(\|V_d \mathbf{1}^n\|_2 - \|V_d - v\|_2, 0)/\|V_d \mathbf{1}^n\|_2 - \alpha \sum_{i \in AV} \max(h_{max} - h_i, 0) \tag{5}$$

where v is a speed vector of all the vehicles in the network, V_d is the desired speed scalar, $\mathbf{1}^n$ is a $\mathbf{1}$ vector with n elements, where n is the total number of vehicles in the network, α is an adjustable constant, h_i is the headway between the ith AV and its leader, and h_{max} is a constant of expected headway.

Average Speed Reward We define an instantaneous systematic average speed reward as

$$r_t = \frac{\sum_{i=1}^{n} v_i}{nV_{max}} \tag{6}$$

where n is the current number of all vehicles in the traffic network, and V_{max} is the maximum speed allowed on every lane. This reward is provided every time step. Summing it over the entire episode and then dividing the sum by the episode's horizon T gives the value of average speed (Equation 1)

of the episode.

Outflow Reward The reward for instantaneous outflow is

$$r_t = O_t \tag{7}$$

where O_t is the number of vehicles that leave the observed area of the traffic network through any lane during the tth time step. We note that the sum of this reward over the simulation will always be proportional to the Outflow metric (Equation 2) by a factor of $1/T$, assuming the simulation occurs over a fixed period of time. Thus optimizing for this reward is equivalent to optimizing for the Outflow metric.

4.3 Modular Transfer Learning Approach

We would like to scale up to the I-696 Merge scenario, which is much larger than scenarios used in past research. As a result, the network can contain many more vehicles than networks used in past work, and also more simulation time before congestion can occur. Learning under this scenario can be problematic for the following three reasons. First, the state and action space exponentially increases as the number of controlled vehicles increases. Specifically, the combined action space of the system is, $|A| = |A_i|^{N_{AV}}$ where $|A_i|$ is the size of the action space of a single AV, and the size of the combined state space is $|S| = |S_i|^{N_{AV}}$ where $|S_i|$ is the size of the state space of a single AV. Second, since the centralized agent can only receive a global reward, and its actions can only have a major impact near the merge point, the reward signal is extremely noisy from activity on the rest of the network. Third, there is an extremely large delay from when the AVs take an action until they receive a reward that's related to the action's impact on the outflows of the network since AVs don't exit the network long after they pass the merge. Therefore, we propose using transfer learning and a modular approach to learn in this scenario.

4.3.1 Method

We create a window surrounding the junction so that the length of each merge component is comparable to the smaller networks which we trained on. We then take a policy that was trained in a small network, and apply it to the AVs inside the window, while outside of the window, AVs act like human drivers. We refer to this approach as the *Zero-Shot Transfer* approach. and compare it to one trained from scratch only within the window. We refer to this approach as *Train from scratch (Window)*.

4.3.2 State and Reward

The states and rewards employed in the modular approach are the same as in the centralized method.

5 Empirical Evaluation

In this section we present our experimental setup for performing an empirical evaluation of methods discussed above.

5.1 Traffic Scenario 1 - The Simple Merge

Our Simple Merge experiments are based off of the Flow benchmark [18]. The road network consists of a main highway of length 600m before the merge and a merging lane of 200m. After merging, the vehicles still need to travel an additional 100m. The junction controller is a "priority" controller where both incoming edges have equal priority. This controller is the same as in previous benchmarks [18]. If two vehicles arrive at the junction at the same time with equal priority, the one with the lower speed will yield to the vehicle with the higher speed. The main highway has an inflow of 2000 vehicles per hour consisting of both humans and AVs. The merging lane has an inflow of 200 vehicles per hour, made up entirely of human drivers. For both inflows, vehicles enter the traffic network with a uniform time gap. In the mixed autonomy traffic flow, the RL vehicles are equally spaced among human vehicles. In the centralized experiment the maximum number of controlled AVs, N_{AV}, is 5.

5.2 Traffic Scenario 2 - The I-696 Merge

The I-696 network has the same shape as the real-world Interstate 696 highway in the US, which is a much larger network than the simple merge. In our experiments we simplified the I-696 network to

have a single rather than multiple lanes, a main road, and a single merging road as highlighted in Figure 1, we refer to this part of the network as the I-696 Merge. The I-696 Merge is much longer than the simple merge, which makes it challenging for existing methods to learn effective driving policies. The highway length before the merge is 3131m, the merging edge length is 1878.56m, and after merging the vehicle will still need to travel 5077.7m. The defined traffic inflows are the same as in the Simple Merge, where the highway inflow is 2000 vehicles/hour containing 10% AVs, and the merge inflow is 200 vehicles/hour made up of human drivers entirely. At the junction in I-696 Merge, the controller is the same as the that in the Simple Merge.

5.3 Human-Driven Vehicles

The movements of human driven vehicles on highways are modelled by the *Intelligent Driver Model* (IDM) [15] that tries to keep a 1-second executed time headway among vehicles.

5.4 Autonomous Vehicles (AV)

Autonomous vehicles are only included in the main highway inflow with a 10% penetration rate and equal spacing. There are at most 5 controlled AVs in Simple Merge and 100 in the I-696 Merge.

5.5 Training Details

All experiments are trained with the same set of parameters using the Proximate Policy Optimization (PPO) algorithm [11]. Both tasks were trained in an episodic manner with a horizon of 2000 time steps of length 0.5 seconds. All results are obtained from SUMO 1.6.0.More details on the training hyper-parameters and MDP parameters are provided in the Appendix. To maintain the anonymity of the authors we do not share our code base at this time, but plan to make it publicly available upon acceptance.

6 Results

All results are obtained from running 100 independent evaluations, which vary in the inter-arrival times of the vehicles entering the network. We've provided 3 selected videos of experiments with the supplementary materials, videos of other experiments will be made available upon request.

6.1 Comparison of Reward Functions

In table 1 we test the simple merge scenario described in Section 5.1, and compare the three reward functions described in Section 6.1 (along with human driven traffic as a baseline), with respect to the three metrics of average speed, average outflow, and average inflow.

Table 1: Statistics of Reward Functions on Simple Merge

$Reward$	Average Outflow (vehs/hr)	Average Inflow (vehs/hr)	Average Speed(m/s)
Human	1559.88±2.758	1726.68±2.611	7.27± 0.029
Original Flow Reward	1690.70±6.131	1746.76±6.339	15.80±0.102
Average Speed Reward	1521.72±3.067	1560.42±4.136	**18.67**±0.106
Outflow Reward	**1801.80**±7.362	**1862.28**±7.181	15.96±0.092

The results are obtained from 100 independent evaluations and we report the mean values of metric readings accompanied with their 95% confidence interval bounds.

All reward functions - the original Flow reward, the average-speed reward, and the outflow reward - improve the average speed over the human baseline. Using instantaneous average speed as a reward results in the highest average speed in the network. However, we see that this improvement comes at the cost of overall reduced network throughput, even when compared with the human baseline. The Average Speed reward produced network inflows and outflows that were significantly lower than the human baseline (an independent T-Test yields p-values < 0.001 for both metrics). By contrast, both the Flow reward function and the outflow reward function are able to increase all 3 metrics, however the Outflow reward function still outperforms the Flow reward. by a statistically significant margin in

all 3 metrics (an independent T-Test yields p-values < 0.001 for both inflows and outflows, and a p-value of 0.024 for average speed).

6.2 Modular Transfer Learning

In this section we compare the performance of three different approaches on the I-696 Merge, and human-driven traffic. The approaches include:

- The Zero-Shot Transfer approach - our proposed modular transfer approach where a policy is trained on Simple Merge and applied to a window in I-696 Merge without additional training.
- The Train from scratch (Window) approach - trained from scratch on a window in I-696 Merge, and applied to the window.
- The Train From Scratch (N_{AV}=100) - trained on the entire I-696 Merge with a maximal number of controlled AV, $N_{AV} = 100$, and applied to up to 100 AVs in I-696 Merge.

All three approaches were trained in two conditions: once with the Flow reward, and once with the Outflow reward. Table 2 demonstrates the success of the Zero-Shot Transfer approach, in combination with the Outflow reward, in terms of the Outflow metric. Training policies directly from scratch reduces all metrics under both rewards. The original Flow reward never beats the human baseline in

Table 2: Transferring a policy from Simple Merge to I-696 Merge

Experiment	Reward	Average Outflow(vehs/hr)	Average Inflow(vehs/hr)	Average Speed(m/s)
Human	None	934.20±6.270	2185.20±0	16.27±0.104
Train From Scratch (N_{AV}=100)	Outflow	437.26±7.134	759.60±5.328	15.11±0.161
	Flow reward	816.12±6.135	1441.84±8.234	16.73±0.112
Train From Scratch (Window)	Outflow	969.19±7.797	2133.40±8.263	16.31±0.118
	Flow reward	925.27±6.196	2087.42±6.588	15.75±0.100
Zero-Shot Transfer (Window)	Outflow	**984.46**±8.493	2065.39±7.119	16.19±0.123
	Flow reward	854.46±12.52	1976.11±11.18	14.76± 0.165

The results are obtained from 100 independent evaluations and we report the mean values of metric readings accompanied with their 95% confidence interval bounds.

terms of outflows, in any of the training approaches. The Train From Scratch in a window approach integrated with the outflow reward can produce better-than-human performance but not as good as the transferred policy (the difference between them is statistically significant with p-value=0.01 in an independent T-Test).

Note that outflows in i696 are approximately half of those in Simple Merge due to the length of i696, vehicles simply take a long time to reach the end of the simulated highway. Since the simulation on I-696 is much slower than on Simple Merge, training on I-696 takes approximately 5 times longer for the same number of iterations than training on Simple Merge.

7 Conclusion and Future Work

In this work we investigate reinforcement learning for traffic control in open networks. We demonstrate that the previously used metric of average speed is an insufficient measurement for open network traffic efficiency, since inflows can be moderated by the agents to achieve higher average speed. To address this, we propose the outflows of the network as a more reliable metric for evaluating traffic efficiency in open networks. We further show that by using the outflows as a reward function, our RL algorithm can generate a driving policy which is superior to a policy generated by the state-of-the art reward function in terms of both outflows, and average speed in a small open network.

After showing that existing methods cannot improve traffic efficiency in large open networks, we develop a modular transfer learning approach which applies the policy learned in the small network to a window surrounding a junction in a large network, specifically the I-696 highway network. Our results indicate that the modular approach achieves better outflows both than human-driven traffic, and than a policy trained from scratch on the full network. On top of the improved traffic efficiency, a key advantage of the transfer learning approach is that it requires much less training time than a policy trained on the entire network, at less than a fifth of the time in our setup.

An interesting avenue for expanding this research in future work is by scaling the modular approach to more than one window.

References

[1] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[2] K. Dresner. Aim: autonomous intersection management. pages 1732–1733, 01 2008.

[3] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 530–537, 2004.

[4] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338, 2016.

[5] K. Jang, E. Vinitsky, B. Chalaki, B. Remer, L. Beaver, A. A. Malikopoulos, and A. Bayen. Simulation to scaled city: zero-shot policy transfer for traffic control via autonomous vehicles. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, pages 291–300, 2019.

[6] N. Kheterpal, K. Parvate, C. Wu, A. Kreidieh, E. Vinitsky, and A. Bayen. Flow: Deep reinforcement learning for control in sumo. *EPiC Series in Engineering*, 2:134–151, 2018.

[7] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[8] 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)*, pages 1475–1480. IEEE, 2018.

[9] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[10] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[12] 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*, 89:205–221, 2018.

[13] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

[15] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

[16] M. Treiber and A. Kesting. Traffic flow dynamics: Data, models and simulation. 2012.

[17] M. Treiber and A. Kesting. The intelligent driver model with stochasticity-new insights into traffic flow oscillations. *Transportation research procedia*, 23:174–187, 2017.

[18] 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*, pages 399–409, 2018.

[19] C. Wu, A. M. Bayen, and A. Mehta. Stabilizing traffic with autonomous vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6012–6018, 2018.

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

[21] H. Youn, M. T. Gastner, and H. Jeong. Price of anarchy in transportation networks: efficiency and optimality control. *Physical review letters*, 101(12):128701, 2008.