Mixed-Autonomy Traffic Control with Proximal Policy Optimization

Haoran Wei, Xuanzhang Liu, Lena Mashayekhy, and Keith Decker
Department of Computer and
Information Sciences
University of Delaware
Newark, DE, USA
Emails: {nancywhr, xzliu, mlena, decker}@udel.edu

***Abstract*—This work studies mixed-autonomy traffic optimization at a network level with Deep Reinforcement Learning (DRL). In mixed-autonomy traffic, a mixture of connected autonomous vehicles (CAVs) and human driving vehicles is present on the roads at the same time. We hypothesize that controlling distributed CAVs at a network level can outperform the individually controlled CAVs. Our goal is to improve traffic fluidity in terms of the vehicle's average velocity and collision avoidance. We propose three distributed learning control policies for CAVs in mixed-autonomy traffic using Proximal Policy Optimization (PPO), a policy gradient DRL method. We conduct the experiments with different traffic settings and CAV penetration rates on the Flow framework, a new open-source microscopic traffic simulator. The experiments show that network-level RL policies for controlling CAVs outperform the individual-level RL policies in terms of the total rewards and the average velocity.**

I. INTRODUCTION

Traffic congestion has always been a significant issue, especially in metropolitan areas. Congestion will be even worse in the near future due to the explosive growth in the number of vehicles and the limited road network expansion. In the United States, 87.9 percent of daily commuters use private vehicles [1]. However, it is hard to alleviate traffic congestion because there are many different factors to consider, such as vehicles changing lanes or collisions. The traffic flow itself is dynamic and stochastic and thus it is hard to capture or observe in real-time (e.g., inconsistent driving speeds may cause stop-and-go waves). Fortunately, new transportation technologies including connected infrastructure and connected vehicles pave the way to more intelligent transportation systems [2]. Mixed-autonomy traffic is a mixture of connected autonomous vehicles (CAV) and human-driven vehicles. Early work showed that using CAVs can improve traffic flow in terms of speed and stability [3], [4], [5]. However, most of the work held the perspective of CAV platooning or an individual CAV. In this study, we extend mixed-autonomy traffic control to a distributed, multi-agent scope.

Deep Reinforcement Learning (DRL) has been used as a powerful tool in solving control problems and has achieved significant success in many complex systems including robotic control [6] and gaming [7]. We believe DRL is a promising approach for solving traffic control as it is a theoretically sequential decision-making problem. Compared to other approaches (e.g., game theory), DRL provides more flexible solutions without high computation cost on the fly. In recent years, RL has made many breakthroughs in the intelligent transportation area, such as self-driving car control [8], [9], coordinated traffic lights [10], and other connected infrastructure systems [11]. Vehicle driving control is a continuous time-sequential task. RL also can be used to optimize the driving velocity or more complex behaviors (e.g., merging). Autonomous vehicles can improve traffic flow and fuel consumption by adjusting their speeds as well as avoid oscillations with human-driven vehicles [5]. Even a small percentage of autonomous vehicles could have a significant impact to potentially reduce total fuel consumption by up to 40 percent and braking events by up to 99 percent [5].

In this study, we consider distributed CAVs within a certain distance as a multi-agent network and use DRL to learn their cooperative driving control policies to optimize mixed traffic flow in terms of improving the average velocity. We propose three network-level control strategies: single-agent asynchronous learning; joint global cooperative learning, and joint local cooperative learning. We use the first one as the baseline to compare with the latter two, which learn a joint global control policy over multiple CAVs. We hypothesize that network-level control can improve the control policy performance compared to individual CAV's independent controls. We use Proximal Policy Optimization (PPO) [12], a DRL method, to learn the CAVs control policies. The experiments are conducted on an open-source framework, Flow [13] with the SUMO built-in environment. The experiment settings include mixed traffic with 10%, 20%, and 30% CAVs, respectively. The experiments show that a network-level RL policy outperforms an individual control RL policy in terms of the total rewards and the average velocity. An RL reward is associated with the current velocity compared to the desired velocity within a safety threshold. The desired velocity is a high velocity that traffic flow is expected to drive at without any safety concerns, and it can be designed by a human expert. In this paper, we use the speed limit as the desired velocity. The total RL rewards are the reward summation within a given time horizon. The average velocity of the traffic flow is a more straightforward metric, and it is proportional to the cumulative rewards. Due to their proportional relationship, it is reasonable to train the RL control policy by maximizing the cumulative rewards which will lead to a high average velocity in the real world.

The rest of the paper is organized as follows. In the next section, we present the existing work in this domain. We then formulate our problem as a multi-agent Markov Decision Process (MDP) model in Section III and provide a brief overview of the traffic control strategies. In Section IV, we propose our three learning strategies for the CAVs control problem. In Section V, we evaluate the properties of the proposed strategies by extensive experiments. Finally, we summarize our results and present possible directions for future research.

II. Related Work

With the rapid growth of autonomous vehicle technologies, it is reasonable to envision near future mixed traffic conditions, where autonomous and human-driven vehicles coexist. In the early years, Game Theory was widely used to build smart traffic systems, including traffic light control [14] and vehicle-to-vehicle interactions [15], [16]. Khanjary [14] employed Cournots oligopoly game to solve the traffic light controlling problem. In the proposed game model, streets were considered as players and competed to increase their share of green light time. From the vehicle aspect, Elhenawy et al. [15] proposed an algorithm inspired from the chicken-game for traffic control at uncontrolled intersections to reduce the average travel time and delay. Different from previous studies that consider traffic lights, Wei et al. [16] extend the traffic controlling problem to the case that there is no explicit traffic signals. They designed a hybrid game strategy for connected autonomous vehicles in order to maximize intersection throughput and to minimize traffic accidents and congestion. However, these approaches may perform poorly in large-scale scenarios due to the inherent high computational complexity of many game-theoretic approaches.

Recently, several studies have focused on equipping autonomous vehicles with RL controllers to alleviate traffic problems in various scenarios, such as traffic light control [17], Vehicle-to-Infrastructures (V2I) network scheduling [11], and vehicle driving control [18], [19]. We categorized these studies into two groups: 1) improving traffic stabilization and 2) improving average velocity (throughput).

Improving traffic stabilization. Traffic flow is a non-stationary system that may produce backward propagation waves in different shapes of roads causing part of the traffic to come to a complete stop [20]. Related autonomous vehicle control strategies have been proposed such as "FollowerStopper" and "PI with Saturation Controller" that aim to reduce the emergence of stop-and-go waves in a traffic network. However, the performance of these approaches are sensitive to the parameters set and limited to a known desired velocity. Wu et al. [13] demonstrates that with DRL methods, using the same state information and samples from the overall traffic system, DRL surpasses the state-of-the-art hand-crafted controllers in terms of system-level velocity. However, the trade-off is larger headways. Kreidieh et al. [18] shows the feasibility of DRL on dissipating these stop-and-go traffic waves in mixed traffic. Similarly, Vinitsky et al. [19] shows the application of RL controllers on more complex road situations such as on-ramp merging.

Improving throughput. Several studies have shown DRL's success in controlling traffic (e.g., in traffic light control). Liu et al. [17] optimized large-scale real-time traffic light control policy with a Deep Q-network (DQN) to increase the system's throughput. The proposed DQN algorithms are tested in a linear topology with several intersections to confirm their ability of learning desirable structural features. Lin et al. [21] utilized the Actor-Critic method to optimize a large-scale traffic light system to maximize the capacity of each traffic road and balance the traffic load around each intersection. Garg et al. [22] proposed a vision-based DRL approach to solve the problem of congestion around the road intersections. They implemented their scheme on a traffic simulator and showed that their method increase the traffic throughput through the intersection in a simple traffic light intersection scenario. Similarly, we explore an approach with DRL to control traffic at a network level. However, we form the network with CAVs only and integrate these two goals to improve the throughput by increasing the traffic flow.

III. Problem Definition

This study considers mixed-autonomy traffic, where multiple connected autonomous vehicles (CAVs) are distributed arbitrarily among human-driven vehicles and drive with Reinforcement Learning (RL) control policies. We model a mixed traffic flow as a discrete-time Markov Decision Process (MDP), defined as $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \rho_0, \mathcal{P}, \mathcal{R} \rangle$, where $\mathcal{N}$ is the model's capacity of the number of agents (CAVs), $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, ρ_0 is the initial state distribution, $\mathcal{P}$ is the transition model, and $\mathcal{R}$ is the reward function. The transition model $\mathcal{P}$ represents the environment dynamics: $\mathcal{P}(s'|s,a) \in [0,1]$ where $s, s' \in S$ and $a \in \mathcal{A}$. The reward function, $\mathcal{R}(s'|s,a) \simeq \mathcal{R}(s') \in \mathbb{R}$, outputs a real number as a reward measuring how good a transition $\langle s, a \rangle \rightarrow s'$ is, and it can be approximated by measuring how good the next state s' is in this scenario. A parameterized RL policy π_θ (θ is is the policy approximator's parameter) outputs a probability distribution across the whole action space. Given an input state, an RL agent will take the action with the highest probability if it is exploited from the RL policy. Let $\eta(\pi_\theta)$ be the discounted total reward following a policy π_θ with a certain time horizon T: $\eta(\pi_\theta) = \mathbb{E}_\tau[\sum_{t=0}^{T-1} \gamma^t r_t]$, where γ is the discount factor, $\tau = (s_0, a_0, \cdots, s_{T-1})$ is the entire trajectory, and each action is selected by the policy π_θ: $s_0 \sim \rho_0(s_0)$, $a_t \sim \pi_\theta(a_t|s_t)$ and $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$. Our goal is to find the optimal policy π^* by maximizing the total reward $\eta(\pi_\theta)$.

In our study, each single-CAV state consists of its absolute speed and its headway. The headway can be calculated by the vehicles' absolute positions. Vehicles' absolute positions and velocities are assumed to be accessible by the V2V and V2I technologies. Actions represent the velocity changes at each time step, and the values are continuous between $[-1, 1]$. The

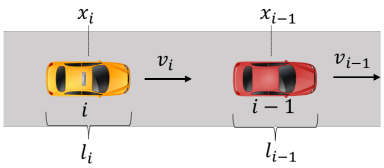


Fig. 1. IDM controller example with vehicles indexed by i and $i-1$: vehicle $i-1$ is the leader of i and i is the follower of $i-1$

traffic flow itself is a continuous process, and we discretize it as a sequence of discrete steps. The state-action-state is set to be deterministic in the experiments. The reward function is predefined based on a desired high velocity and the current velocity aiming to encourage a CAV to drive as fast as possible, while avoiding safety risks. More detailed are presented in Section IV.

We now introduce the control strategies of human-driven vehicles (Intelligent Driver Model) and the RL learning method (Proximal Policy Optimization) for the driving policies of CAVs.

A. Human-driven Vehicle Control

We use the Intelligent Driver Model (IDM) [23], a time-continuous car-following model, to model human driving behaviors and represent the human-driven vehicles' dynamics of the positions and velocities. Considering two adjacent vehicles which are indexed as $i-1$ and i, vehicle $i-1$ is directly in front of i, as shown in Fig 1. Their absolute positions are indicated by x_{i-1} and x_i, respectively, measured from a fixed reference position. The length of vehicle i is len_i. At a certain time step t (we omit the time t in the notation in the following equations for simplification), the acceleration of vehicle i controlled by the IDM controller is represented by a_i^{IDM}:

$$a_i^{IDM} = \frac{dv_i}{dt} = a[1 - (\frac{v_i}{v^*})^\delta - (\frac{s^*(v_i, \Delta v_i)}{s_i})^2], \tag{1}$$

where all notations and parameters are described as:

- $s_i = x_{i-1} - x_i - len_{i-1}$: the headway from vehicle $i-1$,
- v_i: the current velocity of vehicle i,
- s^*: the desired headway which represents the minimum safe distance between two vehicles, formulated as
$$s^*(v_{i,\Delta v_i}) = s_0 + \max(0, v_i T + \frac{v_i \Delta v_i}{2\sqrt{ab}}), \tag{2}$$
where b is the comfortable braking deceleration
- v^*: the desired velocity (velocity in free traffic).

B. Reinforcement Learning

Reinforcement Learning (RL) is a category of machine learning which learns policies for solving sequential decision making problems through interaction with the real environment. The RL policy is optimized by maximizing the cumulative rewards within a time horizon. In autonomous vehicle control problems, the RL decisions work on controlling the vehicle driving dynamics, such as changes in velocity or lane at each discrete time point. In this study, we only consider the decisions on velocity changes, and the actions are assumed to be continuous: a positive value for an acceleration and a negative value for a deceleration. In the traffic control scenario, real environment interactions are infeasible for safety concerns, thus a control policy can be explored within a simulator where vehicles' absolute positions and velocities can be accessed in real time. An RL module as an external part can be connected to the simulator and learn an RL policy with the collected data. The complete workflow is shown in Fig 2. All CAVs are assumed to be homogeneous which means they have the same dynamical features (e.g., acceleration or deceleration response time), routing controller (e.g., controlling algorithm), and reward function.

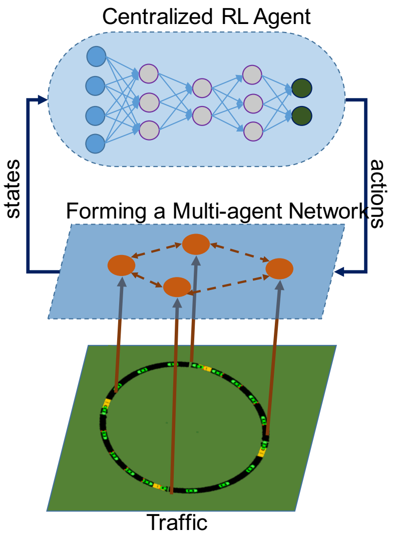


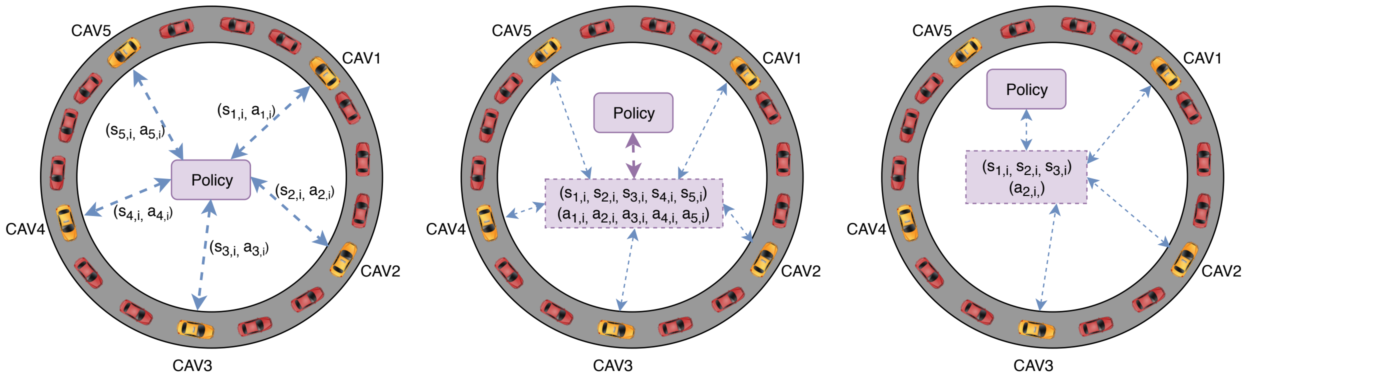
Fig. 2. A multi-agent network can be formed from the distributed CAVs, and an RL controller learns a cooperative control policy to adjust the CAVs driving behaviors (acceleration or deceleration). We use 4 CAVs as an example here.

C. Learning with Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) [12] is a policy-based RL method with significantly less computational complexity than other policy gradient methods. Instead of imposing a hard constraint, PPO formalizes the constraint as a penalty in the objective function and updates the policy directly by maximizing the discounted total reward as:

$$\eta(\pi_\theta) = \mathbb{E}_\tau[\pi_t(a|s;\theta)A_t(s,a)], \tag{3}$$

where $\pi_t(a|s;\theta)$ is the current parameterized policy and θ is the policy's parameter. In addition, $\mathbb{E}_\tau[\cdots]$ indicates the empirical expectation of rewards within a certain time horizon over a finite batch of trajectories, and τ is a CAV driving trajectory. In this study, each trajectory contains 2000 discrete time steps (seconds), or it terminates early if a collision happens. At each time step, a decision on the velocity change will be made by the RL policy. CAVs change their speeds accordingly, then CAVs update their state based on sensory data. The policy is represented as a neural network, where θ represents



(a) Shared single-agent learning (b) Global joint cooperative learning (c) Local joint cooperative learning

Fig. 3. Three policy learning processes with different autonomous vehicle networks (yellow vehicles are autonomous vehicles, and red vehicles are human-driven). Shared single-agent policy is defined based on single-agent states and actions; global joint cooperative policy is updated with the joint states and actions over all agents; local joint cooperative policy regulates a single agent's actions with the local joint states of two adjacent CAVs.

the network's weights, bias and other hyper-parameters. An advantage value $A(s, a)$ is defined for each state and action pair. This value measures how good an action a is compared to the average performance of all actions in a given state s. We use $A(s_t, a_t)$ equivalently with $A_t(s, a)$, and the advantage value for state s_t and action a_t is calculated as:

$$\begin{aligned} A(s_t, a_t) &= Q(s_t, a_t) - V(s_t), \\ Q(s_t, a_t) &= r_t + \sum_{i=1}^{T-1} \gamma^i r_{t+i} + \gamma^{t+T} V(s_{t+T}), \end{aligned} \tag{4}$$

where $Q(s_t, a_t)$ is the estimated discounted total reward the CAV will receive by taking action a_t at state s_t, and $V(s_t)$ is the estimated discount reward from state s_t onwards. Note that T is a time horizon for look ahead, and γ is the discounted factor. In general, a value approximator network can be trained independently from the policy approximator for the value $V(s)$.

Moreover, a policy ratio R_t is defined to evaluate the similarity between the updated policy and the previous policy at time step t as:

$$R_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}(a_t|s_t)}}, \tag{5}$$

where a large value of $R_t(\theta)$ means that there is a large change in the updated policy compared to the old one, $\pi_{\theta_{\text{old}}}$. The policy controls the actions, which are velocity changes, thus a large policy change may cause a large velocity change within one time step and lead to a safety issue. Therefore, to avoid large changes in velocity, we use the clipping of PPO which constrains policy updates within a reasonable range, as follows:

$$\text{clip}(R_t(\theta)) = \begin{cases} R_t(\theta) & \text{if} \quad 1-\epsilon \leq R_t(\theta) \leq 1+\epsilon \\ 1-\epsilon & \text{if} \quad 1-\epsilon > R_t(\theta) \\ 1+\epsilon & \text{if} \quad R_t(\theta) > 1+\epsilon \end{cases} \tag{6}$$

where ϵ is a small positive constant.

With the constrained policy update and clipping operation, the policy optimization objective function can be adapted from Eq (3) as:

$$\eta^{\text{CLIP}}(\pi(\theta)) = \mathbb{E}_\tau[\min(R_t A_t, \text{clip}(R_t), 1-\epsilon, 1+\epsilon)A_t], \tag{7}$$

where A_t abbreviates the advantage value $A_t(s, a)$ at time step t, clip$(\cdot)$ is the clipping function, and R_t is short for $R_t(\theta)$.

When the advantage value A_t is positive, the objective function value is at most $(1+\epsilon)A_t$, because the ratio $R_t(\theta)$ is bounded by $(1+\epsilon)$. On the other hand, when A_t is negative, the objective function value is bounded between $(1-\epsilon)A_t$ and A_t. A set of driving trajectories are collected within a time horizon, and the policy is updated by maximizing the clipped discounted total reward (as Eq (7)) with a gradient ascent.

IV. Methodology

In this study, we propose three different learning strategies with multiple CAVs: a) shared single-agent learning, b) global joint cooperative learning, and c) local joint cooperative learning, as shown in Fig 3.

A. Shared Single-agent Learning

A shared policy, as our baseline, is an individual-level strategy. This policy is learned for single CAV's states and actions, however, it is updated with all CAVs' driving experience data simultaneously, as shown in Fig 3(a). This is a process of updating a centralized single-agent policy by using a decentralized execution. An action is a continuous number within a range representing the speed change at one discrete time step, where a positive value is for acceleration and a negative value is for deceleration. One state $s_i(t) = \{v_{i,t}, h_{i,t}\}$ of CAV i includes the current absolute speed $v_{i,t}$ and the current time headway $h_{i,t}$ between CAV i and CAV $i-1$, where CAV $i-1$ is directly in front of CAV i. A time headway for a vehicle is the duration of time to catch up to the vehicle directly in front without a change in the current speed of vehicles:

Algorithm 1 Shared Single-Agent Policy

```
Initialize policy network with random weighs $\theta_0$ and clipping threshold $\epsilon$
Initialize experienced data buffer $\mathcal{B}$
for episode = $1, \dots, M$ do
    for CAV=$1, \dots, N$ do
        Collect trajectories $\{\tau_i\}$ on policy $\pi(a_{i,t}|s_{i,t};\theta)$
        Extend $\mathcal{B}$ with $\{\tau_i\}$
    end for
end for
Estimate advantage $A$ with Eq (4)
Update the policy by $\theta' \leftarrow \arg\max_\theta \eta^{\text{CLIP}}(\theta)$ as Eq (7)
```

Algorithm 2 Global Joint Cooperative Policy

```
Initialize policy network with random weighs $\theta_0$ and clipping threshold $\epsilon$
for episode = $1, \dots, M$ do
    if $N < \mathcal{N}$ then
        form joint states as $\mathbf{s} = \{s_1, \cdots, s_N, 0, \cdots, 0\}$
    else if then
        from joint states as $\mathbf{s}_0 = \{s_1, \cdots, s_{\mathcal{N}}\}$
    end if
    Collect set of trajectories on policy $a_t \sim \pi(a_t|s_t;\theta)$
    Estimate advantage $A$ with Eq (4)
    Update the policy by $\theta' \leftarrow \arg\max_\theta L^{\text{CLIP}}(\theta)$ as Eq (7)
end for
```

$h_{i,t} = l_{i-1,i}/v_{i,t}$, where $l_{i-1,i}$ is the distance between two adjacent CAVs that is calculated using the difference of their absolute positions. The reward function is defined to optimize the vehicles' velocities while maintaining safety and is adapted from the reward function proposed in [24]:

$$r_{i,t} = \max(\|\hat{v}\| - \|\hat{v} - v_{i,t}\|, 0)/\|\hat{v}\|, \tag{8}$$

where $r_{i,t}$ is the reward for CAV i at time step t, and $\hat{v}$ is the desired velocity, an arbitrary large value to encourage high velocity. The advantage of this strategy is that it collects more information at one time (high data sample efficiency) because all CAVs can use their observation data to update a shared policy in parallel. However, the policy may have high variance or oscillation due to the frequent updates from different CAVs with different aspects of the environment. The shared single-agent learning strategy is summarized in Alg 1.

B. Global Joint Cooperative Learning

In the global joint cooperative learning scenario, the policy is defined with the joint states and joint actions. The joint state space is defined as $\boldsymbol{\mathcal{S}} = \mathcal{S}_0 \times \cdots \times \mathcal{S}_{\mathcal{N}}$ where $\mathcal{N}$ is the system capacity, therefore, all joint states have the same size of $\mathcal{N}$. If there are less than $\mathcal{N}$ CAVs in the system, the joint states are post zero padded. Similarly, the joint action space is defined as the cross product of each CAV action space as well: $\boldsymbol{\mathcal{A}} = \mathcal{A}_0 \times \cdots \times \mathcal{A}_{\mathcal{N}}$.

We provide two reward functions with the max operation presented in Eq (9) and the average operation presented in Eq (10). Both rewards are defined on velocities of all CAVs in the system as follows:

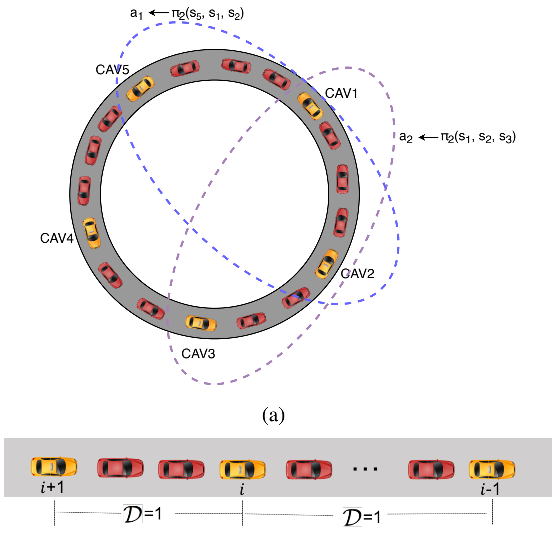
$$r_t = \max_{i \in \{1,\dots,\mathcal{N}\}} (\|\hat{v}\| - \|\hat{v} - v_{i,t}\|, 0)/\|\hat{v}\| \tag{9}$$

$$r_t = \mathbb{E}_{i \in \{1,\dots,\mathcal{N}\}}[\max(\|\hat{v}\| - \|\hat{v} - v_{i,t}\|, 0)]/\|\hat{v}\| \tag{10}$$

A centralized connected infrastructure can be used to learn this centralized policy in this scenario (as shown in Fig 3(b)). Alternatively, real-time information of a single CAV (e.g., velocity and position) can also be sent between pairs of CAVs through vehicle-to-vehicle (V2V) communications so that global joint states can be formed on every single CAV. However, in this situation, the communication cost grows exponentially with the number of CAVs. With PPO as the learning method, the procedure of learning the control policy with a global joint cooperative learning is summarized in Alg 2.

C. Local Joint Cooperative Learning

In order to alleviate the high communication cost with joint global policy, a joint policy in a smaller scale is defined with a local MDP $\langle \mathcal{D}, \boldsymbol{\mathcal{S}}, \mathcal{A}, \rho_0, \mathcal{P}, r \rangle$, where $\mathcal{D}$ is the local network radius of a CAV based on the number of CAVs (as shown in Fig 3(c)). Specifically, CAV i, the $\mathcal{D} - 1$ CAVs in the front of it, and $\mathcal{D} - 1$ CAVs following that CAV compose a local joint CAV network of $2\mathcal{D} - 1$ CAVs. Therefore, the state of CAV i is formed by the states of CAVs in its local network. Note that the policy is defined for a single CAV. For example, Fig 4 shows two local networks $\{\text{CAV}_5, \text{CAV}_1, \text{CAV}_2\}$ and $\{\text{CAV}_1, \text{CAV}_2, \text{CAV}_3\}$. In the first network, CAV_1 is the main learner which adjusts its speed according to the local joint states of CAV_5 and CAV_2. In the second network, CAV_1 is



(a)

(b)

Fig. 4. Demonstration of local joint cooperative learning with $\mathcal{D} = 1$ (Red vehicles are human-driven and yellow ones are CAVs.)

Algorithm 3 Local Joint Cooperative Policy for one agent

```
Initialize policy network with random weighs $\theta_0$ and clipping threshold $\epsilon$
for episode = $1, \dots, D$ do
    Reset experience buffer $\mathcal{B}$
    for t= $0, \dots, T$ do
        Detect neighboring CAVs within radius of $\mathcal{D}$
        Form joint states $s_t = \{s_{i-\mathcal{D}}, \dots, s_i, \dots, s_{i+\mathcal{D}}\}$
        Collect transition $(s_t, a_t, s_{t+1})$ on policy $\pi(a_t|s_t)$
        Extend $\mathcal{B}$ with transitions
    end for
    Estimate advantage $A$ with Eq (4)
    Update the policy by $\theta' \leftarrow \arg\max_\theta L^{\text{CLIP}}(\theta)$ as Eq (7)
end for
```

only a part of the joint state for CAV$_2$ which learns its control policy using its local network. In this example, the radius is 1 for both local networks. Our hypothesize is that a small-scale local joint cooperative CAV network performs better compared to the single-CAV shared policy, and requires less communication cost than the global joint cooperative solution.

V. Experiments

A. Simulator

Our experiments are conducted with Flow [13] which is an open-resource framework for DRL implementation in SUMO [25], a microscope traffic simulator. Flow combines the RL library RLlib [26] (RLlab) in multiple traffic scenarios including ring-shaped roads, traffic light grids, and on-ramp merging. We use the built-in ring-shape scenario in this study. We also utilized Ray [27] to allow multiple CAVs asynchronous updates (in the case of the shared single-agent RL policy). In the simulator, each human-driven vehicle is modeled as an IDM and CAVs are controlled with RL described in Section III.

B. Scenario Setup

Three comparison experiments with different CAV penetration rates are conducted in a ring-shape road shown in Fig 5. We classify the CAV penetration rates as low (10%), medium(20%), and high(30%), where they represent different levels of autonomy. PPO is used as the RL learning method. In a ring-shape road, local slow speed congestion is caused by an individual vehicle's deceleration or inconsistent driving speed. An extreme case is that if a vehicle drives very slowly or stops, the whole traffic flow can suffer a stop as a consequence. This phenomenon behaves like a traveling wave (also called "stop-and-go" wave). With an optimal driving strategy, CAV can drive relatively faster within a safe distance from its neighboring vehicles, and it can lead the following vehicles behind the CAV have a better driving experience and thus it enhances the whole flow driving performance.

Our goal is to provide an autonomous vehicle control strategy at a network level to alleviate the traffic stop-and-go waves and increase the average velocity. PPO is our learning method and the RL policy's performance is evaluated with 1) the average total reward for individual CAVs from each training episode and 2) the average velocity of the whole traffic flow. We set the road length to 230m. CAVs are set near-uniformly distributed among human-driven vehicles. All of the parameters of the experimental setup are summarized in Table I, where a, b, v^*, s^* are the parameters for human-driven vehicle control (IDM controlled discussed in Section III-A.) and the others are for the RL learning. Time horizon is the total discrete time steps in each training episode; in some cases, the episode terminates early due to a collision. We keep these settings the same for all three learning strategies under three different CAV penetration rates. Finally, we only present the reward obtained by Eq (9) for the global joint cooperative learning as we observed similar performance with both reward functions. For simplicity, we use the terms "single-agent policy", "global joint policy", and "local joint policy" to represent the policy learned based on the shared single-agent learning, global joint cooperative learning, and local joint cooperative learning, respectively.

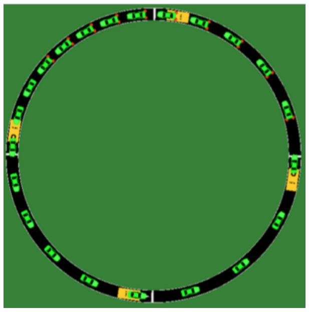


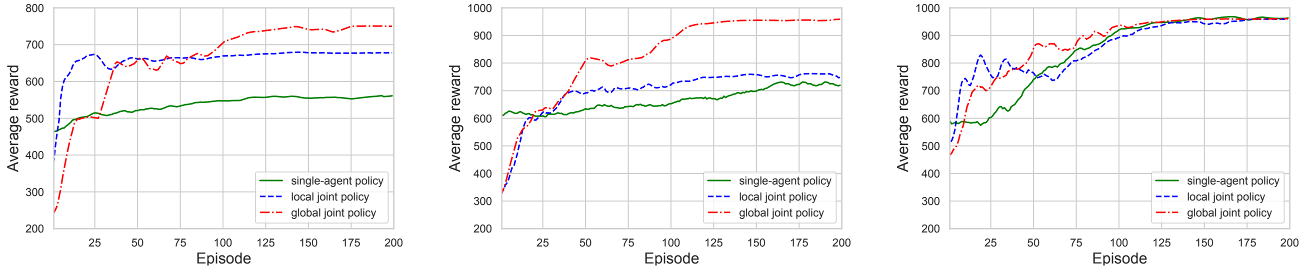
Fig. 5. A Ring-Shape Road

TABLE I
Parameters

Parameter	Setting
horizon	2000
trajectories	20
No. of human-driven	30
No. of CAVs	$3, 6, 9$
a	$1m/s^2$
b	$1.5m/s^2$
v^*	$30m/s$
s^*	$1s$
ϵ	0.3
λ	0.999
$\hat{h}$	$1s$
$\hat{v}$	$25m/s$

C. Performance and Analysis

We run 200 training episodes with each environment setting. Fig 6 shows the average total reward received in each iteration considering different CAV penetration rates. Fig 7 shows the obtained average velocity in each iteration. Overall, all three RL policies perform better with a higher penetration rate of CAVs. Both global policy and the local policy surpass the baseline, shared single-agent policy, in all three different CAV

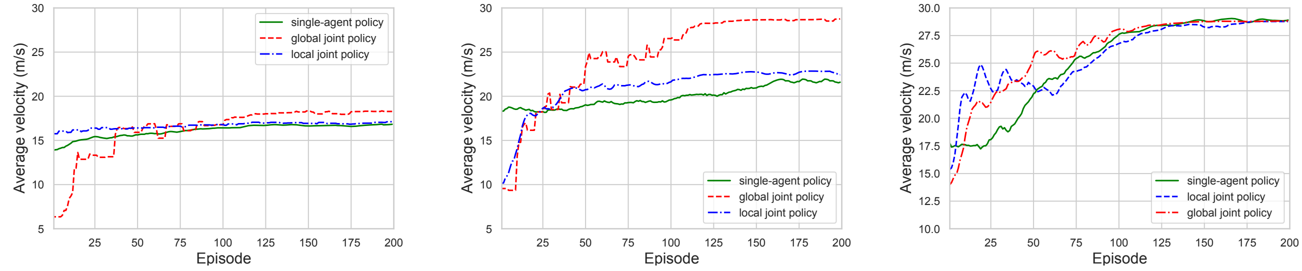


(a) Total reward with 10% CAVs

(b) Total reward with 20% CAVs

(c) Total reward with 30% CAVs

Fig. 6. Comparison of training rewards



(a) Average velocity with 10% CAVs

(b) Average velocity with 20% CAVs

(c) Average velocity with 30% CAVs

Fig. 7. Comparison of average velocities

penetration rates. Both joint RL policies with 30% CAVs converge faster compared with 20% and 10% because these joint policies consider collaboration among the CAVs and capture more information about the environment. The joint global policy achieves the best convergence in terms of total reward and average velocity, especially in 10% and 20% CAVs. Moreover, in a 10% CAV penetration rate, all RL policies perform similarly in the obtained average velocity and the impacts of the proposed joint policies on velocity are significantly higher with a higher CAV penetration rate. This suggests a lower bound on the number of CAVs in the traffic (e.g., $\geq$ 20%) to achieve a high stable speed for all vehicles and guarantee an optimal control performance (reaching the target velocity without accident). For example, with 20% penetration of CAVs, the average velocity obtained by the global joint policy almost reaches the target velocity (30m/s).

With fewer CAVs in the traffic, the RL policies at the network level (both joint global policy and local joint policy) perform $\sim$ 1.3 times better than the baseline, shared single-agent policy. With 30% CAVs, the highest total reward is reached by all three policies, and the fastest average velocity is obtained as well. The results show that with a sufficient rate of CAV penetration, any autonomous control policy, even at the individual level, can influence the traffic flow positively. Moreover, the system has a performance upper bound with a fixed traffic setting by reaching the minimum distance between adjacent vehicles to avoid accidents. Another observation is that a high total reward results in a high average traffic flow velocity under different traffic settings, which means learning the control policy by maximizing the total RL reward can be correctly transferred to the real world and result in a high average traffic flow velocity.

When there are fewer CAVs, the joint global policy does not outperform the joint local policy significantly. However, both of them surpass the shared single-agent policy (e.g., Fig 6(a)). The shared single-agent policy learns slowly with lower CAV penetration rates, and the RL policy performance does not improve over the time. This is caused by frequent updates with insufficient experience data. On average, the local joint policy does not perform as well as the joint global policy due to its limited view. However, with a sufficient CAV penetration rate, the local joint policy has asymptotic performance as the joint global policy.

To investigate the impact of radius on the performance of the local joint cooperative learning, we consider three different radii: $\mathcal{D} = 1$, $\mathcal{D} = 2$, and $\mathcal{D} = 3$ for a CAV and compare its performance in terms of the average total reward along each trajectory. As shown in Fig 8 (data are smoothened with each 5 episodes), a larger radius brings a better performance. This is due to the fact that a more global view or higher number of joint states are captured and thus a better cooperation can be learned.

In summary, considering that the local joint policy requires much less V2V and V2I communications, we believe this policy could be the best choice for the mixed autonomy traffic regularization with a sufficient CAV penetration rate. With a few number of CAVs, however, the joint global policy is the

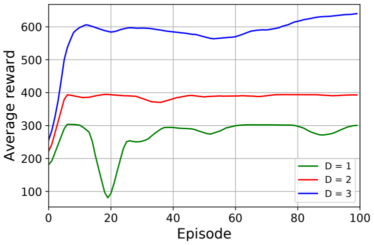


Fig. 8. Average total reward with different local joint network radius ($\mathcal{D}$)

best choice as the communication cost would not be high due to the small number of CAVs.

VI. Conclusion and Future Work

This paper implements deep reinforcement learning in traffic optimization under mixed-autonomy traffic conditions. Compared to the state-of-the-art where *individual* RL controls are solved with reinforcement learning, we proposed *network-level* learning policies for CAVs. Experimental results were conducted on a microscopic traffic simulator (Flow), and the results showed the network-level policies outperform the individual-level policy and the RL policy learned with customized rewards can also be correctly transferred to velocity control. The global joint policy obtains the best performance, however it leads to high communication overhead as the penetration rate of CAVs increases. When there is no available V2I resources or V2V communications are costly, the joint local policy is a better choice. In our future work, we plan to study impacts of communication cost and latency on the control policies, instead of analyzing them intuitively. In addition, we plan to design a more efficient individual-level policy to stabilize the policy updates.

References

[1] A. Downs, "Traffic: Why it's getting worse, what government can do," Brookings Institution, Tech. Rep., 2004.

[2] N. Wang, X. Wang, P. Palacharla, and T. Ikeuchi, "Cooperative autonomous driving for traffic congestion avoidance through vehicle-to-vehicle communications," in *Proc. of the IEEE Vehicular Networking Conference (VNC)*, 2017, pp. 327–330.

[3] S. E. Shladover, "Review of the state of development of advanced vehicle control systems (avcs)," *Vehicle System Dynamics*, vol. 24, no. 6-7, pp. 551–595, 1995.

[4] B. Besselink and K. H. Johansson, "String stability and a delay-based spacing policy for vehicle platoons subject to disturbances," *IEEE Trans. on Automatic Control*, vol. 62, no. 9, pp. 4376–4391, 2017.

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

[6] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[8] T. Schmidt-Dumont and J. H. van Vuuren, "Decentralised reinforcement learning for ramp metering and variable speed limits on highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 8, p. 1, 2015.

[9] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 11, pp. 3204–3217, 2017.

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

[11] R. Atallah, C. Assi, and M. Khabbaz, "Deep reinforcement learning-based scheduling for roadside communication networks," in *Proc. of the 15th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2017, pp. 1–8.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

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

[14] M. Khanjary, "Using game theory to optimize traffic light of an intersection," in *Proc. of the 14th IEEE International Symposium on Computational Intelligence and Informatics*, 2013, pp. 249–253.

[15] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 343–347.

[16] H. Wei, L. Mashayekhy, and J. Papineau, "Intersection management for connected autonomous vehicles: A game theoretic framework," in *Proc. of the 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 583–588.

[17] X.-Y. Liu, Z. Ding, S. Borst, and A. Walid, "Deep reinforcement learning for intelligent transportation systems," *arXiv preprint arXiv:1812.00979*, 2018.

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

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

[20] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecksexperimental evidence for the physical mechanism of the formation of a jam," *New journal of physics*, vol. 10, no. 3, pp. 1–7.

[21] Y. Lin, X. Dai, L. Li, and F.-Y. Wang, "An efficient deep reinforcement learning model for urban traffic control," *arXiv preprint arXiv:1808.01876*, 2018.

[22] D. Garg, M. Chli, and G. Vogiatzis, "Deep reinforcement learning for autonomous traffic light control," in *Proc. of the 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, 2018, pp. 214–218.

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

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

[25] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using SUMO," in *Proc. of the 21st IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 2575–2582. [Online]. Available: https://elib.dlr.de/124092/

[26] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, "Ray rllib: A composable and scalable reinforcement learning library," *arXiv preprint arXiv:1712.09381*, 2017.

[27] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan *et al.*, "Ray: A distributed framework for emerging AI applications," in *Proc. of the 13th USENIX Symposium on Operating Systems Design and Implementation*, 2018, pp. 561–577.