Efficient Connected and Automated Driving System with Multi-agent Graph Reinforcement Learning

Tianyu Shi*, Jiawei Wang*, Yuankai Wu, Luis Miranda-Moreno, Lijun Sun

Abstract—Research on connected and automated vehicles (CAVs) has received considerable attention from both industry and academia in recent years. The advances in reinforcement learning have led to substantial progress in controlling a single CAV. However, in mixed autonomy involving both CAVs and human-driving vehicles, it remains a critical challenge to encourage cooperation among CAVs to better regulate human-driving traffic flow and improve the efficiency of the whole transportation system. In this paper, we address the CAV cooperation problem in mixed autonomy by incorporating Connected Automated Vehicle Graph (CAVG) into multi-agent reinforcement learning (MARL) to model the mutual interplay among CAVs. In the proposed framework, CAV cooperation is first learned using graph convolutional networks with attention mechanism. We conduct extensive experiments based on the car-following and unsignalized intersection settings in *flow*. We also evaluate our method on the merging scenarios — an open road network with a varying number of agents — to demonstrate the generalization ability. Experimental results show that the proposed MARL-CAVG framework outperforms state-of-the-art baselines for CAV control in improving the efficiency and safety of mixed autonomy transportation system.

Index Terms—Connected and Automated Driving, Decision making, Multi-agent Reinforcement Learning, Graph Neural Networks.

I. INTRODUCTION

TRAFFIC system is unstable in nature due to the inherent randomness in human-driving behavior [1]. Shockwaves and stop-and-go have become a primary safety concern and the main driver for traffic accident. As a promising solution to improve the efficiency of transportation systems, connected and automated vehicles (CAVs) have received increasing attention from both industry and academia. One major benefit of CAVs is that the randomness in driving behavior can be significantly reduced; thus, the whole system can be better managed by control algorithms with minimum reaction time. Theoretically, having a fully autonomous fleet will substantially enhance the capacity and efficiency of urban transportation systems. However, before reaching full autonomy, it is inevitable that both CAVs and human-driving vehicles co-exist and interact with each other. In such a mixed-autonomy system, how to

design and optimize CAV behavior becomes critical to the development and implementation of future autonomous driving.

With recent advances in artificial intelligence, reinforcement learning (RL) has become an efficient tool to solve diverse, intelligent control tasks. In particular, RL fits various tasks in autonomous driving, and the field has been greatly advanced thanks to recent development in RL (e.g., flight control [2], Go games [3]). In particular, RL fits various tasks in autonomous driving, and the field has been greatly advanced thanks to recent development in RL (see, e.g., [4], [5]). However, despite these advances, many research challenges remain in promoting CAV cooperation in mixed autonomy. First, as CAVs have different characteristics compared to human-driving agents (e.g., reaction time and action generation process), it becomes challenging to navigate them in such an extremely dynamic and complicated driving environment. Second, in a mixed-autonomy system, it remains unclear how to encourage automated vehicle agents to cooperate to maximize the total expected returns of the whole system. Finally, how to effectively guarantee both safety and efficiency in policy generation is also an urgent research question in a multi-agent automated driving setting.

In this work, we try to address the above challenges by incorporating graph neural networks into a multi-agent reinforcement learning framework to better encourage cooperation. In doing so, we integrate information obtained from each CAV into a graph structure—Connected and Automated Vehicle Graph (CAVG), in which the learned latent features are exploited to enhance cooperation among multiple agents. Our main contributions are:

- To the best of our knowledge, this work is the first to use the graph attention networks to capture mutual interplay among CAVs in the navigation setting, which is beneficial for cooperative control.
- We propose a dynamic adjacency matrix scheme in the decision-making framework to account for both speed and position information, extract valuable information from surrounding neighbors.
- We conduct extensive experiments with diverse levels of complexity. We compare the proposed MARL-CAVG model with several state-of-the-art baselines for model evaluation. Our MARL-CAVG model can better balance between safety and efficiency in mixed-autonomy system.

The rest of the paper is organized as follows. Section II summarizes the related literature, in particular recent developments in RL. Section III introduces the proposed MARL-CAVG model for CAV control. Extensive experiments based on simulated scenarios are presented in Section IV. Section V summarizes

* Contributed equally.

The authors are with the Department of Civil Engineering, McGill University, Montreal, QC H3A 0C3, Canada.

information will be further illustrated and analyzed in experimental studies. Moreover, we consider each agent c_i can communicate with other CAV agents, therefore the overall state observation of the i-th CAV is denoted by $S_i = \{o_i^1, ..., o_i^m\}, i = 1, ..., N$, where m is the maximum number of neighboring CAV agents which c_i can sense. In this way each CAV agent can develop more cooperative policy by incorporating information from neighbors before making control decision.

- **Action**: Action a_i, $i = 1, ..., N$ is the speed adjustment of CAV agents. It is bounded by the maximum allowable acceleration a_{acc} and allowable deceleration a_{dec} (i.e., $a_i \in [a_{dec}, a_{acc}], a_{dec} < 0, a_{acc} > 0$).
- **Reward function**: We define specific reward functions for different simulation scenarios.
  For the *ring* scenario and the *figure eight* scenarios, we define the reward function to encourage high average speeds from all vehicles in the network and penalize accelerations/decelerations by the CAVs. The rewards for *ring* and *eight* scenario are defined as:

$$r_i = -w_v \times (\bar{v}_T - \bar{v}) + w_a \times (\hat{a} - \bar{a}), \quad (1)$$

where w_v, w_a stand for the weight parameters for average velocity and average acceleration which are predefined as 2 and 4 respectively. Note that $\bar{v}_T$ is the target velocity and $\hat{a}$ is the threshold of the acceleration.
  For the *merge* scenario, reward function encourages the consistency between the system-level speed and desired speed, while slightly penalizing short headways among CAVs [5]:

$$r_i^t = -w_v \times (\bar{v}_T - \bar{v}) + w_h \times (\min(\frac{(\bar{h} - t_{min})}{t_{min}}, 0)), \quad (2)$$

where w_v and w_h denotes the weight parameters for average velocity $\bar{v}$ and average headway $\bar{h}$ which are predefined as 1 and 0.1 respectively. In addition, there is a smallest acceptable time headway t_{min}, which is defined as 1 in this study.

- **Termination**: An episode is terminated if the time horizon is reached or a collision happens.

B. Multi-agent Cooperation Within Mixed-autonomy System

In most previous literature (e.g., in [9], [4]), each CAV maintains its own policy in the environment. This is not efficient for controlling a group of CAVs in mixed-autonomy transportation environment due to training complexity. Therefore, we introduce the shared policy into our control framework. In this way, we can not only exploit experience from different CAVs as they encounter different situation, but also develop a common sense to efficiently navigate these vehicles in the mixed-autonomy system.

To better capture neighboring information, we establish a graph attention based multi-agent reinforcement learning architecture as shown in Fig. 2. Specifically, CAVs learn their policies with PPO as the basic optimization scheme to handle continuous action space. The overall architecture is based on the actor-critic algorithm [19] as shown in Fig. 2.
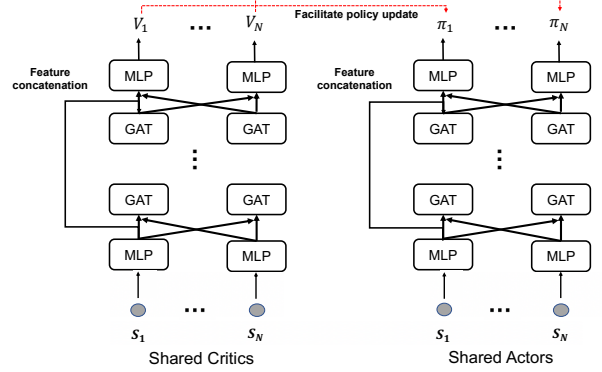
Fig. 2. The architecture of MARL-CAVG. The critic network is a graph convolutional neural network parametrized with ϕ. Notably, the output of critic network at each time step t is state value estimation V_i (i.e., short for $V(S_t, M_i)$), it will be further used for advantage estimation to train the actor network.

In this study, we design and integrate several techniques to encourage cooperation to promote safety and efficiency in dynamic traffic flow:

1) Capture Mutual Interplay among CAVs: In connected and automated driving scenarios, the environment is extremely dynamic because agents keep moving and their relationships among neighbors change quickly. This characteristic makes it very difficult for agents to learn to cooperate with each other.

As shown in Fig. 6, after observing the local state, the CAVs will integrate information from their neighbors to develop a more comprehensive awareness of the current traffic dynamic. Firstly, unlike previous approaches (e.g., [16], [20]), we build the adjacency matrix based on Gaussian speed field using Gaussian process regression (GPR) model [21]. Compared to only using relative position or velocity information, the advantage of using GPR is to capture the spatial and temporal interactions among surround agents and adaptable to the dynamic changing environment. The standard exponential kernel function is computed as:

$$K(x_i, x_j) = A \cdot \exp\left(-\frac{(x_i - x_j)^2}{2\sigma^2}\right), \quad (3)$$

where A is an amplitude constant. x_i represents the position of the ego vehicle and x_j represents the position of the surrounding j^{th} vehicle. And σ is length scale constant controlling how the correlations are decaying with respect to the distance. A small σ indicates fast decay rate, which imposes less correlation on two points that are far away. In our research, we fixed the length scale as 4 meters as suggested by [21].

Furthermore, we can dynamically construct the feature representation in the adjacency matrix at time step t, the posterior distribution of the relative velocity $\Delta V(x_i, x_j)$ between the i-th CAV agent and its surrounding vehicle j at position x_i can be used to model the interaction relationship among vehicles [21]. These elements are defined as follows:

$$M_t(i, j) = K(x_i, x_j)K(x_i, x_j)^{-1}\Delta V(x_i, x_j), dist(x_i, x_j) \le SC_i, \quad (4)$$

We incorporate the location information (i.e., $K(x_i, x_j)$) and velocity difference (i.e., $\Delta V(x_i, x_j)$) of every two agents in the suggested adjacency matrix. Intuitively, the i-th CAV agent will be more sensitive to a closer surrounding vehicle with higher relative speed than a more distant one with lower relative speed (see Fig. 1).

Intuitively, the observation and extracted features of each agent are integrated through graph convolution based on the weighted adjacency matrix M_t:

$$h_i^k = f\left(\text{concat}\left[M_t H^{k-1}, D_i^{-1} M_t H^{k-1}\right] W_i\right), \quad (5)$$

where f is the activation function, D is the degree matrix and h_i^k denotes extracted feature by agent i at the k th layer, which depends on the current adjacency matrix M_t as well as the feature of its neighbors extracted from previous layer $H^{k-1} = \left[h_1^{k-1}, ..., h_N^{k-1}\right]$.

Furthermore, an attention module is added to capture the impact of the surrounding agents. The neighbors are selected within the scan scale SC_i for each CAV individually. Considering $\mathcal{N}_i$ neighbors of ego CAV i, the attention based feature selection on neighboring CAV j can be computed as:

$$q_i = f^{\text{query}}\left(h_i * W^{\text{query}}\right), \quad (6)$$

$$k_j = f^{\text{key}}\left(h_j * W^{\text{key}}\right), \quad j \in \mathcal{N}_i \quad (7)$$

$$v_j = f^{\text{value}}\left(h_j * W^{\text{value}}\right), \quad j \in \mathcal{N}_i \quad (8)$$

$$\phi_{ij} = \text{softmax}\left(\frac{q_i * k_j^T}{\sum_{l \in \mathcal{N}_i} q_i * k_l^T}\right), \quad (9)$$

$$h_i' = \sum_{j \in \mathcal{N}_i} \phi_{ij} * v_j. \quad (10)$$

Note that we omit the layer index here for simplicity. $\mathcal{N}_i$ denotes the set of agent i as well as its selected neighbors. We use f^{query} and f^{key} to encode input features as query-key pairs, then we implement softmax on dot-product between query and key vectors, with which we can quantify the attention score ϕ_{ij} of agent i on neighboring agent j [22]. Based on the value v_j and ϕ_{ij}, new feature vector h_i' for agent i based on its neighbors can be derived by Eq. 10. With this attention scheme, an CAV agent can further utilize information from neighboring CAVs selectively, and thus the framework can promote more effective cooperation.

2) Continuous Action Generation via Proximal Policy Optimization: In a typical reinforcement learning problem, an agent takes action $a \in A$ based on the current state S and acquires the reward R. Unlike previous tasks based on DQN (e.g., in Go games [3]), the CAVs need to generate continuous action space for smooth and efficient control strategy. Therefore, we use Proximal Policy Optimization (PPO) [23] for CAVs to handle continuous action space. We design the critic network as a graph convolutional neural network parameterized by ϕ. Notably, the output of critic network at each time step t is state value estimation V_i (i.e., short for $V(S_t, M_i)$), it will be further used for advantage estimation to train the actor network.

The update of gradient for critic is based on Temporal-Difference (TD) learning [19], which can be formulated as:

$$\nabla_\phi L(\phi) = \nabla_\phi E\left[\sum_{n=1}^{N} (r_i^t + \hat{V}(S_{t+1}, M_i) - \hat{V}(S_t, M_i))^2\right]. \quad (11)$$

The policy π_i (i.e., short for $\pi(a_i|S, M_i)$) can be modelled as a distribution (i.e., Gaussian distribution for continuous control) and also parameterized through the graph convolutional network with parameters θ. Therefore at given time step t, the policy gradient can be derived with the advantage $\hat{A}_i^t, i = 1, .., N$ from critic:

$$\nabla_\theta J(\theta) = \nabla_\theta E_{\pi_{\theta old}}\left[\sum_{i=1}^{N} \min\left(r_i^t(\theta)\hat{A}_i^t, \right.\right.$$
$$\left.\left. \text{clip}\left(r_i^t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i^t\right)\right], \quad (12)$$

where the likelihood ratio $r_i(\theta) = \frac{\pi_{\theta_i}(a_i|S_t, M_t)}{\pi_{\theta old}(a_i|S_t, M_t)}$, and this is done by defining the policy loss function to be the minimum between the standard surrogate loss and an ϵ clipped parameter. It should be pointed out that, for model simplicity, the adjacency matrices are kept the same for the next state value prediction. This assumption makes sense since the variation is limited between two consecutive state observations, especially when the experiment is studied in fine granularity (i.e., simulation resolution is less than 1 s). In addition, we perform on-policy roll-out to collect the experience (i.e., $\{o_i, a_i, r_i\}$) and the advantage estimation for agent i at time step T is calculated as: $\hat{A}_i^T = \sum_t^T \gamma^t r_i^t - \hat{V}(S_t, M_t)$. The overall training algorithm is summarized in Algorithm 1. Note that we adopt the centralized training and decentralized execution framework. We assume that each automated vehicle will have a centralized platform (e.g., cloud platform) which can share information among automated vehicles. And each automated vehicle will have its own control strategy based on local observation during execution.

IV. EXPERIMENTS

A. Experiments Setup

We conduct experiments in Flow[1], an open-source project that supports mixed-autonomy control. Our algorithm is evaluated based on benchmark *car-following* [17], *intersection* and *merge* [24] as shown in Fig. 3, which are common intelligent traffic control scenarios. In the simulation, horizon represents the number of steps per roll-outs, and each time step is 0.1s.

- **Car-following control**. This is a common scenario in the highway without bottleneck. To simplify the training process, we consider a ring-shape network with single lane which is shown in Fig. 3-(a). In the initial condition, all the vehicles are uniformly distributed on the circular

Algorithm 1 Training algorithm for CAVs control based on traffic simulation
___
Set time horizon T steps for each simulation, set scan scale SC for all the agents.
Initialize memory buffer $\mathcal{B} = \emptyset$, batch size as b.
Initialize parameters ϕ, θ for critic and actor networks.
for each episode **do**
  for $t = 1$ to T **do**
    Obtain local state observation $o_i, i = 1,...,N$ and global observation $S = \{o_1,..,o_N\}$.
    for CAV $i = 1,...,N$ **do**
      Obtain adjacency matrix M_i based on given scan scale SC.
      Sample action a_i from $\pi^{old}(S, M_i|\theta)$ to control CAV i.
    end for
    Obtain next local state observation $o_i', i = 1,...,N$ and global observation $S' = \{o_1',..,o_N'\}$ as well as the reward signal r_i for each CAV .
    $\mathcal{B} \leftarrow \mathcal{B} \cup (a_i, o_i, o_i', r_i, M_i)_{i=1}^N$
    if $|\mathcal{B}|\%b = 0$ **then**
      Fetch experience from M and perform roll-out.
      Update ϕ based on Eq. 11
      Update θ based on Eq. 12
      $\mathcal{B} = \emptyset$
    end if
    if collision happens **then**
      Break
    end if
  end for
end for=0
___

road with the same initial speed. Experimental results in [25] show that the system is very unstable, even a tiny fluctuation can grow, and eventually breaks up the homogeneous movement, resulting in a traffic jam.

- **Intersection control**. This is a common urban traffic scenario. In this case, CAV control can help improve the overall efficiency of urban transportation systems. We consider a simple intersection with a figure-eight shape network with one or two circular tracks. (see Fig. 3-(b)).
- **Merge**. This scenario is common in highway networks. Vehicles move from the on-ramp create backwards propagating stop-and-go waves. As a result, perturbations will propagate upstream from the merge point and reduce the throughput of vehicles in the network. For the merge scenario, total number of vehicles is considered as the number of vehicles per hour coming into the highway lane. See Fig. 3-(c) for the implemented merge scenario.

B. Algorithms Setup

We compare our MARL-CAVG method with several state-of-the-art baselines, including not only reinforcement learning frameworks (single-agent and multi-agent) but also car-following models from traffic flow theory. For all experiments, we run 100 episodes with a collection of the average results of

10 random seeds. The explanations for selecting these baselines are given as follows.

Intelligent driver model (IDM):

- Intelligent driver model (IDM) [18] is a commonly used adaptive cruise control method for vehicles that automatically adjusts the acceleration based on distance and velocity information to maintain a safe distance from the leading vehicle. IDM is commonly used to model human-driving behavior in traffic simulators. (e.g., [4, 9]). We consider to add 0.2 random noise on the action to model the uncertainty of human-driving behavior.

Single-agent methods:

- Deep Deterministic Policy Gradient (DDPG) [26]: DDPG is a deterministic version of model-free RL algorithm to deal with continuous action space. CAV agent can reliably learn the optimal policy with continuous actions. We construct a single-agent training framework based on DDPG method, which is similar to [27].
- Proximal Policy Optimization (PPO) [23]: PPO is a gradient-based RL algorithm to deal with continuous action space. Unlike DDPG, PPO is an on-policy algorithm.

Multi-agent methods:

- Multi-agent Deep Deterministic Policy Gradient (MADDPG) [28]: This is a widely used multi-agent framework with centralized critic and decentralized actor. This is a baseline model which does not specifically consider information from neighbors with graph attention mechanism.
- Multi-agent Proximal Policy Optimization (MAPPO): This framework is developed based on the single-agent version of PPO. Unlike in single-agent PPO, different agents will have a shared policy in MAPPO.

C. Performance Comparison

We conduct several experiments in these three networks, the simulation recordings can be found here [2].

Evaluation in Car-Following Control. Fig. 4 shows the training performance of different methods in the car-following control scenario. As we can see, the MARL-CAVG method outperforms other methods with a large margin. We can see from Table I that MARL-CAVG achieves the second highest velocity and the smallest acceleration, which makes it achieve highest return in this scenario. It indicates that our model can better mitigate the shock-wave during the car following control.

(a) Ring network   (b) Figure eight network   (c) Merge network

■ Automated vehicle agent   □ Unobserved humans

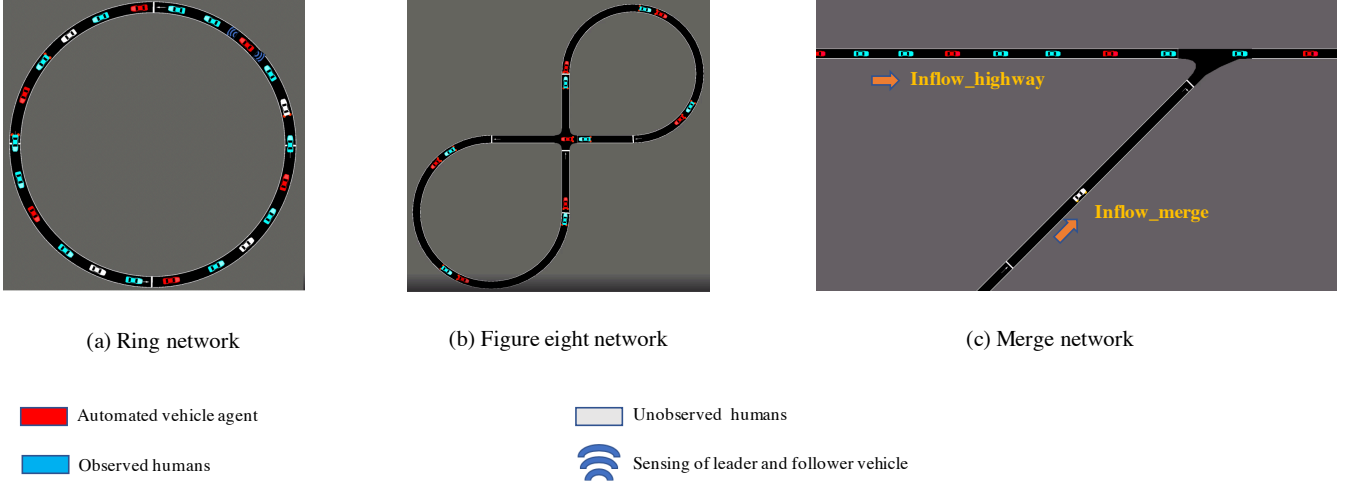■ Observed humans   📶 Sensing of leader and follower vehicle

Fig. 3. Road network structures. To be specific, we consider several variants of simulation scenarios to test the performance of our model. Firstly, for ring network, it's a standard car-following evaluation scenario which is common in the real world. Secondly, for the figure eight network, it's a more challenging scenario compared to the ring network since it has an intersection. Thirdly, for merge network, which is an open-looped network. As a result, it can be used to test the robustness of our method in a dynamic changing environment.
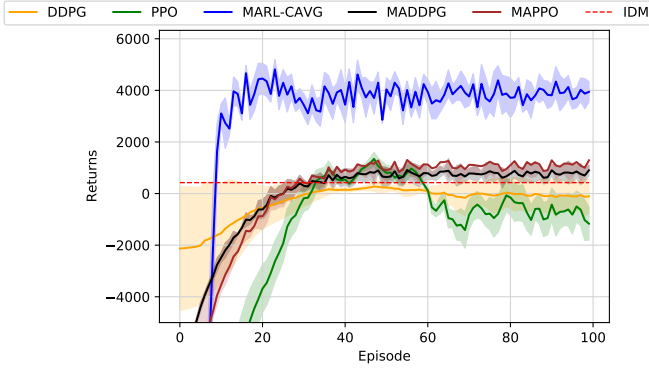

Fig. 4. Learning curves in ring network. In ring network, we use 3000 horizon, the total vehicles are 22 and autonomous vehicles are 16. The lane is 1 and target speed is 30km/h.

a good balance between safety and efficiency. From Table II, we can find that although MARL-CAVG does not achieve the highest velocity, it has the smallest acceleration in this scenario. This demonstrates that our model can learn to sacrifice the speed but achieve higher safety to achieve the optimal trade-off, which is beneficial to get the highest cumulative return.
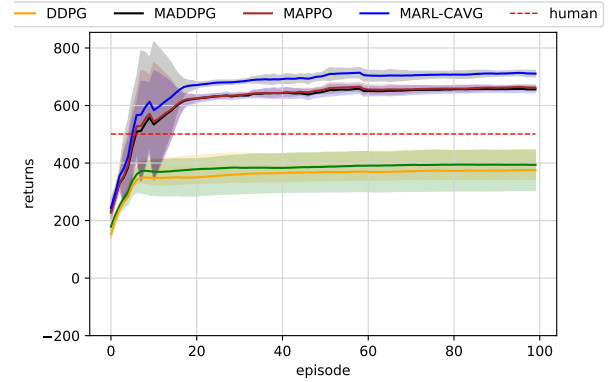
Fig. 5. Learning curves in figure eight network. In figure eight network, we use 1500 horizon, the total vehicle number is 14, and the autonomous vehicle number is 7, and the target speed is 30km/h.

TABLE I
PERFORMANCE COMPARISON IN RING NETWORK

Methods	Velocity (m/s)	Acc (0.1 m^2/s)	Return
IDM	2.754	3.318	424.12
DDPG	3.134 ($\pm$0.148)	2.718($\pm$0.378)	-70.063($\pm$70.506)
PPO	3.165 ($\pm$0.145)	3.129 ($\pm$0.369)	-676.959($\pm$65.046)
MADDPG	3.270 ($\pm$0.148)	2.121 ($\pm$0.366)	779.140($\pm$29.178)
MAPPO	3.379 ($\pm$0.142)	2.782 ($\pm$0.325)	776.225($\pm$20.121)
MARL-CAVG	**3.391** ($\pm$0.092)	**0.835** ($\pm$0.171)	**2593.99**($\pm$51.820)

Evaluation in Intersection Control. Secondly, in the *eight* scenario, the objective is to balance the safety and efficiency in the figure eight network. To be specific, with the introduction of lane change behaviors or increasing target speed, the average speed within the network will increase and therefore the possibility for collisions at the intersection will be higher. As shown in Fig. 5, we find that MARL-CAVG method can achieve better performance when considers the aforementioned changes. It can achieve better control performance, maintaining

TABLE II
PERFORMANCE COMPARISON IN FIGURE EIGHT NETWORK

Methods	Velocity (m/s)	Acc (0.1 m^2/s)	Return
IDM	4.531	9.205	500.87
DDPG	4.325($\pm$2.091)	8.619($\pm$3.191)	379.061(($\pm$46.852)
PPO	4.064($\pm$2.415)	8.812($\pm$4.090)	357.946($\pm$64.509)
MADDPG	**4.879**($\pm$1.231)	6.192($\pm$2.213)	618.641($\pm$32.796)
MAPPO	4.875($\pm$1.312)	6.819 ($\pm$2.131)	619.89($\pm$24.139)
MARL-CAVG	4.265($\pm$1.913)	**3.123**($\pm$1.139)	**669.119**($\pm$28.895)

Evaluation in Merge Control In the *merge* scenario, the

number of controlled and uncontrolled vehicles varies with time due to the inflow and outflow. MARL-CAVG method treats it through a limited multi-agent setting, transforming a tremendous state space using graph attention mechanism to handle the varying feature vector size. In this case, we can also find that our model outperforms the baselines in most evaluation indicators.
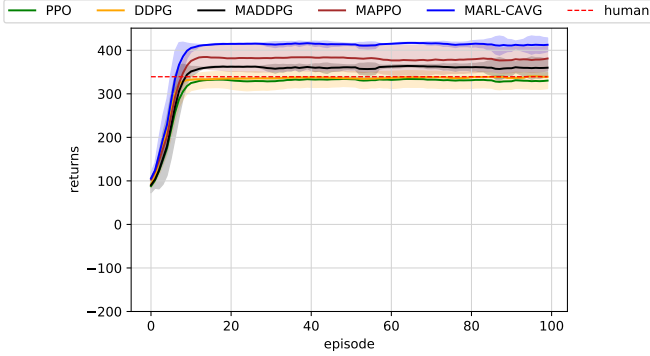
Fig. 6. Learning curves in merge network. We use 600 horizon, we consider the penetration rate as 0.25, i.e., 25% of the vehicles are autonomous vehicles, the number of lane is 1, the target speed is 30km/h.

TABLE III
PERFORMANCE COMPARISON IN MERGE NETWORK

Methods	Velocity (m/s)	Acc (0.1 m^2/s)	Return
IDM	2.881	6.386	339.24
DDPG	4.512(±1.021)	3.87 (±1.013)	318.327(±18.149)
PPO	4.663(±2.183)	3.88(±1.216)	321.415(±15.291)
MADDPG	4.718±1.132)	4.123(±1.516)	325.659(±12.261)
MAPPO	**4.723(±1.912)**	3.298(±1.158)	365.732(±12.261)
MARL-CAVG	4.719(±1.294)	**2.213(±1.391)**	398.220(±9.5703)

### D. Further discussion

1) Visualization of Control Performance: To evaluate the control performance, we select the *ring* network as an example and plot space-time diagram and velocity figures with the trained policy after 200 episodes. The number of head in attention module is 8. We test each method with 200 time steps and a target speed of 20km/h, then record the average speed for all the vehicles in the current road network.

From the result in Fig. 7, we can see that after automation turns on, the velocity will become stable. Furthermore, our model can reach the highest speed compared to other baselines.

To visualize the impact of shock-wave, we further compare the space-time diagram performance in the ring network before and after the automation is turned on. We can see from Fig. 8-(b) that the velocity fluctuates sharply. With automation turned on, the velocity becomes smooth, and the average velocity increases, as shown in Fig. 8-(a).

These findings can demonstrate that while all the models can learn to regulate the traffic, the MARL-CAVG model can achieve the overall best performance.
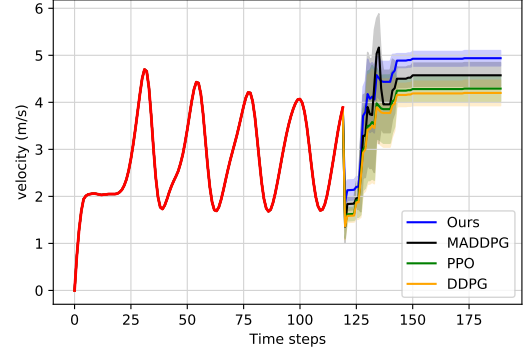
Fig. 7. Velocity performance in ring network. The red curve stands for the control performance with all human driving vehicles which is unstable. After automation turned on, the flow becomes stable.
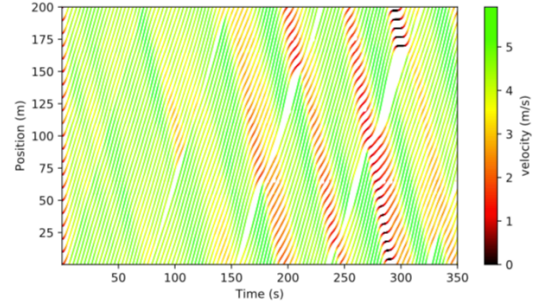
(a) Space time diagram with automation in ring network.
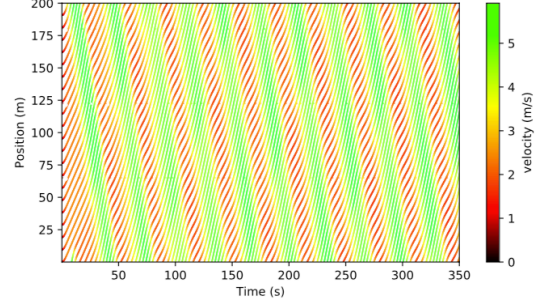
(b) Space time diagram without automation in ring network.

Fig. 8. Space time diagram with and without automation in the ring network. Different colors stand for different velocities. We can see that without automation, the velocity is not smooth, which is due to the stop and go effect, while with automation turned on, the automated vehicles can mitigate shock-wave.

2) Evaluation of different penetration rates: The penetration rate is a critical parameter that can affect the model's performance. To evaluate the performance of our model under different penetration rates, we select the typical ring scenario to make the comparison. We select the typical multi-agent RL approach (MADDPG) and single-agent RL approach (DDPG) as the baselines. As shown in Fig. 9, we can see that with the increase of penetration rates, the return of both the single-agent and multi-agent approaches first increases then decreases. The reason is that with more automated vehicles in the road network, there is a larger control policy space to explore, which hinders the training efficiency. Owing to the parameter sharing and

graph attention within a certain scan scale, the MARL-CAVG can efficiently handle the increased number of controlled agents, and therefore achieve increasing return and the best overall performance.

Fig. 9. Evaluation of returns under different penetration rates (PRs) in ring scenario. We fix the parameter of experiment in Section IV-D1 and only change different PRs.

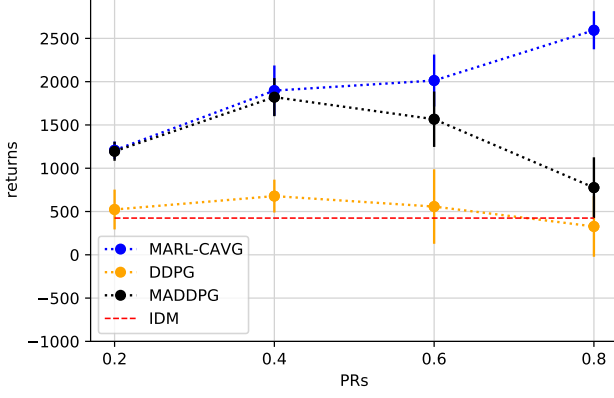*3) Evaluation of different target speeds:* In real-world scenarios, different roads have different speed limits. Under some advanced freeway management frameworks, the speed limits can change over time [29]. In our study, we train the agents with a fixed target speed (20 km/h). It is curious to explore whether the agents can generalize to the scenarios with different speed limits (e.g., whether agents trained on urban roads generalizable to freeways). In this study, we fix the penetration rate as 0.4, and try with different target speeds, then evaluate different methods' performance. We set the target speed 20km/h during training and calculate the percentage(%) of reward increase for each method based on their 20km/h baseline. As shown in Fig. 10, we can find that for each method, the increase of target velocity leads to increase of collected reward. However, when the target velocity is too high (e.g., 120km/h), then the agent's performance will decrease. Our model achieves the best return given the highest target velocity. This demonstrates that our model is more generalizable to scenarios with different speed limits.

4) Evaluation of different architectures: For different architectures of the model, we first evaluate the effects of the range of scan scales on model performance. The results are shown in Figure 11. From the results, we can find that if we slightly enlarge the scan scale, the performance will be better because it can include more neighboring information. However, further increase of scan scale will decrease the model's performance because more redundant information will make learning becomes harder.
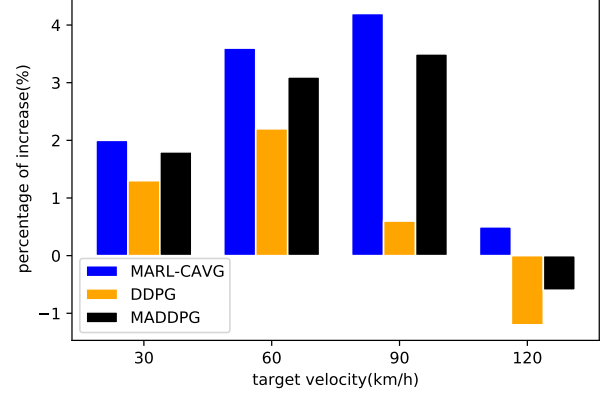
Fig. 10. Evaluation of percentage increase of returns under different target speeds in ring scenario. We fix the parameter of experiment in Section IV-D1 and only change different target speeds. We select the 20km/h as the baseline for each method and try with different target speeds.
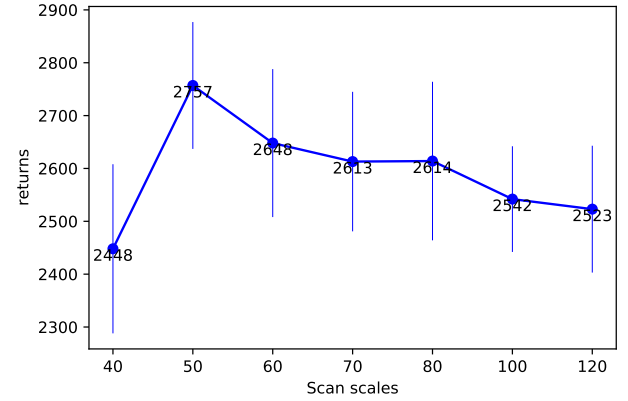
Fig. 11. Evaluation of returns under different scan scales(SCs) in ring scenario. We fix the parameter of experiment in Section IV-D1 and only change different SCs.

Furthermore, we evaluate the performance in terms of different information used to build the adjacency matrix. We consider only speed information, position information, and both speed and position information as given in Equation 4. The different information considered are as follows:

Only consider relative position information:

$$m_i = x_i - x_j, \tag{13}$$

where x_i and x_j are the position of the i-th agent and its surrounding j-th agent.

Only consider velocity information:

$$m_i = \frac{v_T}{v_i(|v_j - v_i|) + \epsilon}, \tag{14}$$

where v_i and v_j are the velocity of i-th agent and its surrounding j-th agent, v_T is the target speed, and ϵ is a small positive number to keep it from divided by 0.

The philosophy of designing these information representations is, if two vehicles are running slowly on the road and are far away from each other, their correlation should be weak so that the measure should be larger. On the contrary, if a vehicle is running fast along with a slow vehicle and they are very closed, then they are more likely to be affected by each other, either because of the safety or efficiency consideration. Therefore, the measure will be smaller, and priority will be higher.

TABLE IV
RETURNS WITH DIFFERENT INFORMATION IN ADJACENCY INDEX

Adjacency index.	position	velocity	both
Returns	2490.99 ($\pm$ 20.149)	2601.87 ($\pm$ 19.825)	2710.32 ($\pm$ 23.581)

We fixed the penetration rate as 0.4, target speed as 30km/h. As shown in Table IV, speed information is more important than position information. Integrating both position and velocity information through velocity field can achieve the best overall performance.

5) Evaluation of attention module: To evaluate the effectiveness of the attention setting, we conduct experiments with/without the attention module and a different number of heads in the attention module. In the experiment, the penetration rate is 0.4; the target speed is 30km/h.

TABLE V
RETURNS WITH DIFFERENT HEADS IN ATTENTION MODULE

Heads	Returns
0	2423.19 ($\pm$39.193)
2	2515.89 ($\pm$48.131)
4	2566.23 ($\pm$43.123)
6	2586.23 ($\pm$41.641)
8	2624.20 ($\pm$41.213)
10	2516.10 ($\pm$39.142)

From Table V, we can find that without attention module, the performance of the model decreases a lot. With the increase of attention heads, it will increase model's performance, while too many heads (heads$\geq$10) will decrease the performance.

V. CONCLUSIONS AND DISCUSSION

In this paper, we propose a graph convolutional reinforcement learning approach for CAV control by encouraging cooperation for efficient traffic control in mixed autonomy.

We conduct extensive experiments based on different road networks and demonstrate the superior performance of our proposed MARL-CAVG method over both reinforcement learning and existing traffic flow simulation baselines. There are two major findings worth noting. First, we find that multi-agent training with the shared policy can achieve much better performance than those single-agent training strategies. Second, efficient communication strategies, such as the graph attention on surrounding neighbors proposed in this paper, can significantly enhance the cooperation among agents, which improves both efficiency and safety of the system. Overall, our method can achieve the best performance under different road networks, target speeds, penetration rates. Our findings provide valuable insights into the design of the connected and automated driving system.

There are several directions for future research and improvements. In particular, as multi-agent training is quite unstable, a small change in the environment setting will result in a large return shift. Thus, it is critical to explore how to better stabilize the training in dynamic settings. In the future, we will also try to develop sim-to-real transfer learning [30] for mixed-autonomy control and implement our approach in real mobile robot vehicles.

REFERENCES

[1] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.
[2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in neural information processing systems*, 2007, pp. 1–8.
[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.
[4] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1379–1384.
[5] 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.
[6] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2016.
[7] Y. Luo, G. Yang, M. Xu, Z. Qin, and K. Li, "Cooperative lane-change maneuver for multiple automated vehicles on a highway," *Automotive Innovation*, vol. 2, no. 3, pp. 157–168, 2019.
[8] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using sumo," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.
[9] T. Shi, P. Wang, X. Cheng, C.-Y. Chan, and D. Huang, "Driving decision and control for autonomous lane change based on deep reinforcement learning," *arXiv preprint arXiv:1904.10171*, 2019.
[10] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.
[11] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," *arXiv preprint arXiv:1911.04175*, 2019.
[12] G. Wang, J. Hu, Z. Li, and L. Li, "Harmonious lane changing via deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, no. accepted, 2020.
[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[14] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International Conference on Machine Learning*, 2019, pp. 2961–2970.
[15] A. Agarwal, S. Kumar, and K. Sycara, "Learning transferable cooperative behavior in multi-agent teams," *arXiv preprint arXiv:1906.01202*, 2019.
[16] J. Jiang, C. Dun, and Z. Lu, "Graph convolutional reinforcement learning for multi-agent cooperation," *arXiv preprint arXiv:1810.09202*, vol. 2, no. 3, 2018.
[17] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1475–1480.

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

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

[20] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1913–1922.

[21] C. Zhang, J. Zhu, W. Wang, and J. Xi, "Spatiotemporal learning of multivehicle interaction patterns in lane-change scenarios," *arXiv preprint arXiv:2003.00759*, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

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

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

[25] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, p. 033001, 2008.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[27] W. Huang, F. Braghin, and S. Arrigoni, "Autonomous vehicle driving via deep deterministic policy gradient," in *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2019.

[28] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.

[29] Y. Wu, H. Tan, L. Qin, and B. Ran, "Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm," *Transportation research part C: emerging technologies*, vol. 117, p. 102649, 2020.

[30] 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*, 2019, pp. 291–300.

Jiawei Wang received the B.S. degree in Traffic Engineering from Sun Yat-Sen University, Guangzhou, China, in 2016, and M.S. degree in Traffic information and control from Sun Yat-Sen University, Guangzhou, China, in 2019. He is now PhD candidate supervised by Lijun Sun, with Department of Civil Engineering at McGill University. His current research centers on Intelligent transportation systems, traffic control and machine learning.

Yuankai Wu received the PhD's degree from the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China, in 2019. He was a visit PhD student with Department of Civil & Environmental Engineering, University of Wisconsin-Madison from Nov. 2016 to Nov, 2017. He is a Postdoc researcher with Department of Civil Engineering at McGill University, supported by the Institute For Data Valorization (IVADO). His research interests include intelligent transportation systems, intelligent energy management and machine learning.

Luis Miranda-Moreno received the Ph.D. degree in transportation engineering from the University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the Department of Civil Engineering, McGill University. He is a specialist in transportation engineering, in the area of intelligent transportation systems, road safety, and the impact of transportation on the environment. He has extensive experience in the integration of emerging technologies and road safety methods for network screening, emerging surrogate safety analysis, and non-motorized transportation. He has worked on various projects for major cities, industries, and government agencies across North America. He has authored over 80 peer-reviewed journal articles.

Tianyu Shi received the B.S. degree in Vehicle Engineering from Beijing Institute of Technology Beijing, China, in 2019. He was a research assistant with Berkeley Deep Drive, University of California Berkeley. He is now pursing master degree supervised by Lijun Sun, with Department of Civil Engineering at McGill University, supported by IVADO excellence scholarship program. His current research centers on reinforcement learning, robotics and intelligent transportation systems.

Lijun Sun received the B.S. degree in Civil Engineering from Tsinghua University, Beijing, China, in 2011, and Ph.D. degree in Civil Engineering (Transportation) from National University of Singapore in 2015. He is currently an Assistant Professor with the Department of Civil Engineering at McGill University, Montreal, Quebec, Canada. His research centers on intelligent transportation systems, machine learning, spatiotemporal modeling, travel behavior, and agent-based simulation.