



Graph Reinforcement Learning for Intelligent Transportation System Control

Tianyu Shi

Department of Civil Engineering

McGill University, Montreal

April 2021

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Engineering

© Tianyu Shi 2021

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.

Abstract

Recent advances in intelligent transportation systems have shown increasing interest in connected and automated-vehicle (CAV) and intelligent traffic-signal controls. Both CAV and signal controls have faced problems because of the complexity of traffic systems. This thesis focuses on studying the aforementioned two control problems, with a particular interest on how to encourage agent cooperation and to improve system robustness by leveraging advances in deep graph reinforcement learning.

The first part of this thesis studies the CAV cooperation problem in mixed autonomy. We introduce Connected Automated Vehicle Graph (CAVG) to multi-agent reinforcement learning (MARL) to model the mutual interplay among CAVs. In this framework, CAV cooperation is learned by using graph convolutional networks and shared policy, and CAV cooperation is further enhanced by introducing attention mechanisms over the graph convolution features. To the best of our knowledge, this work is the first system-level multi-agent cooperative driving framework with graph information sharing. To evaluate the proposed approach, various experiments are conducted in car-following and un-signalized intersection settings. To demonstrate the generalization ability, the proposed method is also evaluated within an open road network (the merging setting) with a dynamic number of agents. Results demonstrate that the proposed MARL-CAVG framework outperforms the state-of-the-art baselines for CAV control and improves performance/efficiency for both CAVs and human-driving.

The second part of the thesis focuses on improving the robustness of large-scale traffic signal control policy to sensor failures and demand surge. We solve this problem by introducing a novel decentralized approach based on graph neural networks (GNN) and distributional reinforcement learning (DRL). Specifically, we model agents as nodes in the graph. We follow a similar approach as in part one to learn the detailed feature representation of each traffic control participant. Furthermore, implicit quantile networks (IQN) are also used to model the state-action return distribution with quantile regression to stabilize the learning of policy. These two objectives are combined together through the loss function concept to improve the overall robustness of our model when dealing with uncertainty. Numerous experiments are also conducted to compare our

approach with existing multi-agent reinforcement learning and transportation approaches. The proposed method can be more robust given missing values and demand surge than other baseline methods.

Keywords: Intelligent Transportation System, Connected and Automated Driving, Intelligent Traffic Signal Control, Decision Making, Multi-agent Reinforcement Learning, Graph Neural Networks.

Résumé

Un système de transport intelligent est une application avancée dans laquelle le contrôle des véhicules connectés et automatisés et le contrôle intelligent des feux de signalisation ont reçu une attention considérable dans l'industrie et le monde universitaire. Ils ont posé des problèmes en raison de la complexité des systèmes de trafic. Cette thèse étudie la manière d'encourager la coopération et d'améliorer encore la robustesse du système basé sur l'apprentissage par renforcement de graphe profond.

Dans la première partie de cette thèse, le problème de coopération CAV en autonomie mixte est étudié en introduisant le graphe de véhicule automatisé connecté (CAVG) à l'apprentissage par renforcement multi-agents (MARL) pour modéliser l'interaction mutuelle entre les CAV. La coopération CAV est apprise à l'aide de réseaux convolutifs de graphes et d'une politique partagée, et la coopération CAV est encore améliorée en introduisant des mécanismes d'attention sur les fonctionnalités de convolution de graphes. À notre connaissance, cette étude est le premier cadre de conduite coopérative multi-agents au niveau du système avec partage d'informations graphiques. Des expériences approfondies sont menées dans des paramètres d'intersection suiveurs de voitures et non signalés. Pour démontrer la capacité de généralisation, la méthode proposée est également évaluée dans un réseau routier ouvert avec des numéros d'agents dynamiques - le paramètre de fusion. Les résultats ont démontré que le cadre MARL-CAVG proposé surpasse les lignes de base de pointe pour le contrôle des CAV et améliore les performances / efficacité à la fois pour les CAV et la conduite humaine.

Sur la base du cadre d'apprentissage par renforcement de graphes, dans la deuxième partie, on étudie la manière d'améliorer encore la robustesse de la politique de contrôle des feux de circulation face aux pannes de capteurs et à la surtension. Une nouvelle approche décentralisée est introduite pour le contrôle des feux de circulation à grande échelle basé sur les réseaux neuronaux graphiques (GNN) et l'apprentissage par renforcement distributionnel (DRL). Plus précisément, les agents participants sont modélisés comme des nœuds dans le graphique. Une approche basée sur l'apprentissage par renforcement graphique dans la première partie est utilisée pour apprendre une représentation détaillée des caractéristiques de chaque participant au contrôle de la circulation.

En outre, les réseaux quantiles implicites (IQN) sont également utilisés pour modéliser la distribution de retour état-action avec régression quantile pour stabiliser l'apprentissage de la politique. Ces deux objectifs sont combinés via la fonction de perte pour améliorer la robustesse globale de notre modèle compte tenu de l'incertitude. De nombreuses expériences ont été menées pour comparer notre approche aux approches d'apprentissage par renforcement multi-agents et de transport. La méthode proposée peut être plus robuste compte tenu des valeurs manquantes et de l'augmentation de la demande que les autres méthodes de référence.

Mots clés: Système de transport intelligent, conduite connectée et automatisée, contrôle intelligent des feux de circulation, prise de décision, apprentissage par renforcement multi-agents, réseaux de neurones graphiques.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Lijun Sun for his guidance. I would like to express my thanks to my co-supervisor Prof. Luis Miranda-Moreno for recommending me for the IVADO scholarship.

I would also like to thank Prof. Laurent Charlin, Prof. Denis Larocque from Mila for their help and guidance on my research on improving the robustness for the traffic signal control project. I would like to thank Mr. François-Xavier Devailly for helping me review the code, analyzing the experiments.

I would like to thank Mr. Jiawei Wang for helping me design the experiments and analyze the algorithm in the connected automated vehicle research. I would like to thank Dr. Yuankai Wu for helping me revise the paper.

I would like to thank Dr. Jie Fu for discussing reinforcement learning algorithms and providing me new ideas. I would like to thank Mrs. Zainab Almheiri for helping me check the grammar and translate my thesis abstract into French.

I would like to thank Dingyi Zhuang and Chengyuan Zhang for discussing course related questions with me. Special thanks to Fuqiang Liu, Xudong Wang, Zhanhong Cheng, Xiao Xu Chen, Ce Zhang, Zhenyuan Ma, Lulu Tan, Mojdeh Shariafi from McGill Smart Transportation group who have helped me throughout my studies at McGill.

Contribution of Authors

This thesis is a manuscript-based report consisting of the following journal/conference manuscripts. Details of the publications are presented below. I would like to declare that I am the sole author of this thesis. My contributions to this research include designing the method, conducting the studies, analyzing the dataset, building the decision models, and writing the manuscript. My supervisor, Prof. Lijun Sun, provided guidance, comments, and editorial revisions throughout the entire process. The co-authors of the publications help me analyzing experiments, providing suggestions, reviewing code, and revising the papers.

(*indicate equal contribution)

Tianyu Shi, Jiawei Wang, Yuankai Wu, Luis Miranda-Moreno, Lijun Sun., 2021. Efficient Connected and Automated Driving System with Multi-agent graph Reinforcement Learning. Transportation Research Board (TRB) 100th annual meeting.

Tianyu Shi, François-Xavier Devailly, Denis Larocque, Laurent Charlin, Lijun Sun., 2021. DGRL: Distributional Graph Reinforcement Learning for Improving Robustness of Large-Scale Traffic Signal Control. The 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'21) (In submission).

Tianyu Shi*, Jiawei Wang*, Yuankai Wu, Luis Miranda-Moreno, Lijun Sun., 2020. Multi-agent Graph Reinforcement Learning for Connected Automated Driving. Workshop on AI for autonomous driving on 2020 International Conference on Machine Learning (ICML).

Table of Contents

Abstract.....	ii
Résumé	iv
Acknowledgements.....	vi
Contribution of Authors.....	vii
Table of Contents.....	viii
List of Tables	x
List of Figures.....	xi
Chapter 1. Introduction	1
1.1. Research background.....	1
1.2. Mixed autonomy system for traffic management.....	2
1.3. Intelligent traffic signal controls.....	3
1.4. Connection between these two problems	4
1.5. Challenges in intelligent transportation system research.....	5
1.5.1. Dynamic and complicated environment	5
1.5.2. Robustness to exogenous uncertainties	5
1.6. Thesis overview and contributions	6
Chapter 2. Literature Review	8
2.1. Multi-agent system	8
2.2. Automated vehicle decision-making and control	9
2.3. Cooperation in multi-agent system.....	10
2.4. Traffic signal control	11
2.5. Robustness of decision making system	12
Chapter 3. Efficient Connected and Automated Driving.....	14
3.1. Mixed autonomy control problem formulation	14
3.2. Methodology.....	16
3.2.1. Multi-agent cooperation within mixed-autonomy system.....	16
3.3. Experiment analysis.....	21
3.3.1. Experiment setup	21
3.3.2. Algorithm setup	23
3.3.3. Performance comparison.....	24
3.3.4. Further analysis.....	28
3.4. Main findings	35
Chapter 4. Robust Large-scale Traffic Signal Control	36
4.1. Traffic signal control problem formulation	36
4.2. Methodology.....	37
4.2.1. Overview of learning process	37
4.2.2. Graph representation learning on different nodes	38
4.2.3. Decentralized distributional RL for TSC.....	40
4.2.4. Multi-objective robust loss	41

4.3.	Experiment analysis.....	42
4.3.1.	Background and assumptions	43
4.3.2.	Experiment setup	44
4.3.3.	Performance comparison	46
4.4.	Main findings.....	55
Chapter 5.	Conclusion and Future Work.....	56
5.1.	Conclusions.....	56
5.2.	Future work.....	57
References.....		58
Appendix.....		64

List of Tables

Table 3.1: Training algorithm for CAV control.....	20
Table 3.2: Performance comparison in ring network.....	25
Table 3.3: Performance comparison in figure-eight network	27
Table 3.4: Returns with different information in adjacency matrix.....	34
Table 3.5: Returns with different heads in attention module.....	34
Table 4.1: Comparison under different traffic regime	47
Table 4.2: Comparison under different missing values in Manhattan network	49
Table 4.3: Comparison under different missing values in Luxembourg network	50
Table A.1: Configuration of different road networks	65

List of Figures

Figure 3.1: CAV control framework.....	14
Figure 3.2: Architecture of MARL-CAVG	17
Figure 3.3: Road network structures	23
Figure 3.4: Learning curve in ring network	25
Figure 3.5: Learning curves in figure-eight network	26
Figure 3.6: Learning curves in merge network	28
Figure 3.7: Velocity performance in ring network	29
Figure 3.8: Space time diagram without automation	29
Figure 3.9: Space time diagram with automation	30
Figure 3.10: Evaluation of different penetration rates	31
Figure 3.11: Evaluation of increase of returns under different target speeds	32
Figure 3.12: Evaluation of different scan scales	33
Figure 4.1: Reinforcement learning for TSC with exogenous uncertainty	36
Figure 4.2: DGRL framework overview	38
Figure 4.3: Visualization of graph representation feature	42
Figure 4.4: Learning scheme for the proposed model	45
Figure 4.5: Trip duration comparison	51
Figure 4.6: Average delays evolution in Manhattan road network.....	52
Figure 4.7: Average delays evolution in Luxembourg road network	52
Figure 4.8: Different κ combination	53
Figure 4.9: Different number of samples	54
Figure 4.10: Comparison of convergence	54
Figure A.1: Traffic demand evolution in Luxembourg road network	64
Figure A.2: Traffic demand evolution in Manhattan road network.....	65

Chapter 1. Introduction

1.1. Research background

As the number of vehicles on our roads keeps rising, it is imperative to adapt traffic conditions to minimize congestion in order to improve transportation efficiency. As solution to the transportation issues, intelligent transportation systems (ITS) have gained momentum. ITS integrates different fields of the transportation system, such as traffic management, automated vehicle control, infrastructure, together to reduce traffic accident risk, traffic congestions, pollution, and satisfy travelers.

One of the important topics in ITS in the last years has been the development of decision-making methods for both autonomous vehicles and traffic controls (Wang et al., 2018; Wu et al., 2017; Wei et al., 2018). Nowadays, most decision making methods are based on predefined plan such as potential field method and model predictive control for autonomous vehicle control (Rasekhipour, Y. et al., 2017; Kim, B. et al. 2001; Ji, J., Khajepour. et al., 2017). For intelligent traffic signal controls, one can refer to fixed-time signal approach (Koonce et al., 2008) and MaxPressure signal approach (Varaiya et al., 2013). However, as in a real-world scenario, some irrational and unseen behaviors may make the predefined plan inefficient.

With recent advances in machine learning, reinforcement learning (RL) has become an efficient tool to model diverse and complex tasks. In particular, RL fits various tasks in intelligent transportation systems, and the field has been greatly advanced thanks to recent developments in RL. In the automated vehicle field, Wang et al. (2019) propose deep reinforcement learning with quadratic networks to generate continuous control actions. Shi et al. (2020) develop a hierarchical reinforcement learning framework for lane change decision making. In the intelligent traffic signal control field, Wei, et al. (2018) test deep reinforcement learning on real-world traffic data and demonstrate superior performance over the predefined plan. However, these single-agent approaches cannot generalize into large-scale networks due to a large joint action space (Wei et al., 2019).

Recent research progress on graph neural networks has enabled effective and scalable reinforcement in multi-agent systems, i.e., mixed-autonomy system and traffic signal control

system. Graph reinforcement learning is proposed to capture multi-agent interplay in order to encourage cooperation (Jiang et al., 2018).

This thesis is based on the graph reinforcement learning framework and integrates it into these ITS problems (CAV control and TSC control). Furthermore, several new findings on cooperation and robustness are also discussed in this thesis.

1.2. Mixed autonomy system for traffic management

The road transportation networks are unstable due to the inherent randomness in human-driving behavior (Treiber et al., 2013). Shock-wave and stop-and-go have become a primary safety concern and the main driver for traffic congestion. As a promising solution to improve the efficiency of transportation systems, connected and automated vehicles (CAVs) have received increasing attention in both industry and academia. One major benefit of CAVs is that the randomness in driving behaviors can be significantly reduce. Thus, the whole system can be better controlled with algorithms, reduce gap times between vehicles and minimum reaction times. 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 exist and interact with each other. Understanding and optimizing CAV behaviors in such a mixed-autonomy road environment is critical to the development and implementation of future autonomous driving.

In particular, RL can help in various tasks in autonomous driving, and the field has been greatly advanced thanks to recent development in RL (see, e.g., Wang et al., 2018; Wu et al., 2017). However, despite these advances, the impact of mixed autonomy is still not fully understood. There are still several challenges to be addressed to obtain the benefits of cooperative automated driving. First, as CAVs have different characteristics compared to human-driving agents (e.g., reaction time and action generation process), it becomes challenging to navigate in such an extremely dynamic and complicated driving environment. Second, in such a mixed-autonomy system, it remains unclear how to encourage automated vehicle agents to cooperate and to maximize the total expected returns of the whole system. Finally, how to effectively guarantee both safety and

efficiency from the policy point of view is also an urgent research question in a multi-agent automated driving setting.

1.3. Intelligent traffic signal controls

Traffic signal controls (TSCs) are a crucial part of a modern ITS environment. TSC can now leverage massive traffic data collected by road and vehicle sensors. As the number of cars on our roads keeps rising, it is imperative to adapt road networks to minimize congestion and reduce its negative impacts (crashes, injuries, emissions, etc.). Developing robust and adaptable traffic control strategies is a powerful mitigating approach as demonstrated in the past (Wei et al., 2018; Devailly et al., 2020; Wei et al., 2019). TSC methods attempt to learn how to adapt traffic signal timing/phasing from available historical and real-time data, including vehicle information such as their positions and velocities (Shi et al., 2019; Essa et al., 2020; Wei et al., 2019).

Such data are often collected from road and on-board vehicle sensors and then transmitted to traffic management centers to help take optimal decisions (e.g., to dynamically change signal indication in a busier lane from red to green). However, missing values in the collected data --- e.g., caused by sensor occlusions and transmission delays -- are a common problem. Missing data can introduce uncertainty in the predictions of the system, which will affect decision-making. Furthermore, traffic demand surge created by events such as roadblocks and incidents will also lead to different congestion situations. Overall, these exogenous uncertainties require robust control policies.

Various simple approaches for TSCs have been proposed such as the classical fixed-time approach (Koonce et al., 2008, Urbanik et al. 2015), which defines a fixed cycle length and phase time for each intersection based on different road conditions. MaxPressure approach (Varaiya et al., 2013) maximizes the throughput of the road networks, i.e., greedily chooses the phase which can maximize the pressure. Due to some unrealistic assumptions, such that the lanes have unlimited capacity and that the traffic flow is constant, their applications in complex real-world scenarios are limited (Varaiya et al., 2013).

Recently, reinforcement learning (RL) has allowed the development of more advanced policies for various traffic control problems (Wei et al., 2018, Wei et al., 2019, Chu et al., 2019). In In general terms, in the RL approach, traffic signal agents take the state input from the environment (road network) and learn to predict the traffic signal phasing and timing (e.g., red/green). The goal of a reinforcement learning agent is to maximize the total return. Traditionally, vehicle delay and/or queue length are seen as measures of travel efficiency to be improved. In particular, RL has been applied to real-world road networks and demonstrated its superiority over other classical control methods such as the fixed-time approach (Koonce et al., 2008, Wei et al., 2018).

1.4. Connection between these two problems

Many cities are moving towards ITS solutions. With the development of 5G technology, vehicles and vehicles will allow to communicate with each other (V2V), and vehicle and infrastructure (e.g., traffic signal controls) are also allowed to share information (V2I). It is expected that the mixed autonomy system and traffic signal control system can work together to provide sustainable mobility for our future. Their connections and similarities are:

1) Common propose

For mixed autonomy system, the objective is to mitigate congestion and penalize inappropriate or dangerous behaviors such as sharp acceleration or deceleration. For traffic signal control systems, the objective is to reduce queue length and mitigate pollution emissions, for instance. As a result, the general objective of these two complementary systems is to increase travel efficiency, to improve safety and reduce emissions.

2) Similar multi-agent settings

For mixed autonomy system, the agent is defined as each connected vehicle. Each vehicle will have its own local observation and generate its own policy to cooperate with other vehicles. For traffic signal control system, each traffic signal controller will also have its own observation and learn to generate its own control phasing strategies for each intersection. As a result, both

systems involve the interaction of multiple agents and the learning to achieve a common objective by cooperating with each other.

1.5. Challenges in intelligent transportation system research

1.5.1. Dynamic and complicated environment

The ITS system, as mentioned above, involves various components, such as vehicles, pedestrians, traffic signal controllers, sensors, etc. Take the mixed autonomy system as an example, when there is a gap in front of the adjacent line of an autonomous vehicle, if the autonomous vehicle makes an abrupt lane change, the surrounding vehicle in the adjacent line could also be affected being forced to decrease its speed sharply. It can end up in a shock-wave in traffic flow. Instead, if the autonomous vehicle learns to cooperate with other agents, adjusts its speed steadily and tries to mitigate the negative impact on the whole system.

On the other hand, for the traffic signal control system, it's a very challenging task to allow the same policy to be optimal for all different road networks. For example, in the Manhattan road network, there are lots of regular shape roads (grid-like), while in Luxembourg, the road network is not a regular shape. Therefore, it is impossible to design and implement the same traffic control strategy and make it applicable for all different road networks.

1.5.2. Robustness to exogenous uncertainties

Modern ITS operations still have uncertainties, not only related to internal components (such as those coming from contain sensors) but also those uncertainties coming from outside the system. As mentioned before, a good decision needs to leverage massive traffic data collected by road and vehicle sensors. However, sensor failures creating missing-data challenges. In addition, various traffic demands will also affect the system's performance.

Take intelligent traffic signal control as an example, for small-scale traffic signal control, reinforcement learning approaches have shown to be robust to demand surge and sensor failure

problems (Rodrigues et al., 2019, Zhang et al., 2020). However, when the system becomes more complicated, i.e., consider more traffic signal controllers in the system or evaluate the model in large-scale network. The model is not robust enough (Wei et al., 2019).

1.6. Thesis overview and contributions

This thesis aims to introduce and develop algorithms that focus on **mixed autonomy control** and **intelligent traffic signal control** in ITS research. This thesis follows this structure. Chapter 2 reviews the literature on multi-agent systems, from general multi-agent systems to specific challenges in ITS research. Then, the thesis considers two typical scenarios, i.e., a mixed autonomy system in Chapter 3 and a traffic signal control system in Chapter 4. In Chapter 3, the thesis is focusing on multi-agent cooperation in mixed autonomy system. In Chapter 4, the thesis provides a deeper analysis of the robustness of the decision-making system.

In this thesis, these challenges are trying to be solved:

1) Cooperation in multi-agent system

Chapter 3, *Research on mixed autonomy system*, focuses on encourage multi-agent cooperation. This thesis uses the graph attention networks to capture mutual interplay in the navigation setting of multi-agent reinforcement learning for mixed-autonomy cooperation. In the mixed autonomy setting, this thesis proposes to integrate a dynamic adjacency matrix scheme in the decision-making framework to exploit both speed and position information from important neighbors and can extract valuable information from surrounding agents.

2) Robustness to uncertainty

Chapter 4, *robustness of intelligent traffic signal control system*, focuses on improving decision robustness to exogenous uncertainty, i.e., sensor failures and demand surge. Guided by previous work, this thesis models each traffic control participants (e.g., TSC, lane, vehicle, connection) as nodes in the graph using graph convolutional networks (GCNs). This thesis shows that using graph representation can implicitly improve the robustness given exogenous uncertainty through better

utilization of the information. This thesis also shows that modeling the distributional state-action value function can explicitly improve learning stability (robust to outliers and converge faster) in the multi-agent decentralized control problem. This thesis further analyzes the trade-off between the aforementioned representation capacity and learning stability. This thesis proposes the distributional graph reinforcement learning approach (DGRL) to strike a flexible trade-off to improve the overall decision performance and system robustness.

Chapter 2. Literature Review

In this chapter, several topics are going to be reviewed. Firstly, in Chapter 2.1, the multi-agent system is reviewed because these two topics (CAV control and TSC control) in this thesis are the multi-agent system. Secondly, in the CAV control part, this thesis focuses on encourage cooperation in the multi-agent system, which is discussed in Chapter 2.2 and Chapter 2.3. On the other hand, in the TSC control part, this thesis focuses on improving robustness, which is discussed in Chapter 2.4 and Chapter 2.5.

2.1. Multi-agent system

Multi-agent system is usually defined as a system which composed of multiple interacting intelligent agents (Hu et al., 2020). Multi-agent system can deal with problems that are hard or impossible for a single-agent system. The typical settings for multi-agent systems are: (1) Fully cooperative, (2) Fully competitive, (3) Mixed cooperative & competitive (4) self-interested (Yang and Wang, 2020). In this context (mixed autonomy and traffic signal control), this thesis considers the fully cooperative relationship among each agent. The difficulty in multi-agent system research is that all the agents' policies cannot remain the same. If all the other agent's policies remain the same, the i^{th} agent cannot get better expected return by changing its own policy because the other agent's objective will change, and therefore, they will change their policy. Thus, the single-agent method cannot be applicable in multi-agent setting.

Typically, there are few ways for multi-agent system control (CAV or TSC control), i.e., fully centralized, fully decentralized, centralized training with decentralized execution (Yang and Wang, 2020).

- 1) Fully decentralized: every agent uses its own observations and rewards to learn its policy.
- 2) Fully centralized: the agents send all information to the central controller. The controller makes decisions for all the agents.

- 3) Centralized training with decentralized execution: A central controller is used during training. The centralized controller is disabled after training. During the execution, the agent will adopt the decentralized fashion. (e.g., Lowe et al., 2017)

In this thesis, the fully decentralized model is adopted. The reason is that it's more common in real setting that each agent (autonomous vehicle or traffic signal controller) will only observe local information rather than full information. To enable better model performance, this thesis considers parameter sharing among each agent, which means that the agents are exchangeable. As a result, the learning would be easier with fewer model to be trained. Furthermore, it would be helpful to tackle cooperation tasks in such multi-agent setting.

2.2. Automated vehicle decision-making and control

In Chapter 3, this thesis is focusing on automated vehicle decision-making and control. Several specific solutions to decision-making for the automated vehicle from both individual and system levels will be illustrated. The cooperation in a multi-agent system will also be further analyzed.

Most existing research in the field of automated vehicles control and motion planning has focused on maximizing the efficiency for an individual agent (i.e., ego driving), which formulates automated motion planning as an optimization problem and solve it with rule-based models (see, e.g., Rasekhipour et al., 2016, Luo et al., 2019). However, such methods may fail in real-life scenarios due to the complex interactions among agents. To address the limitation, recent developments for automated driving have been shifted from rule-based methods to reinforcement learning, which offers more flexibility, efficiency, and superior generalization power. Meanwhile, integration of micro-traffic simulator SUMO (Lopez et al., 2018) with deep reinforcement learning library can enable easy implementation of different traffic control tasks, e.g., lane change, ramp merge, and intersection (Wu et al., 2017). Despite the promising results, this reinforcement learning-based approach still mainly focuses on the control of a single-agent in a static or fully observed setting. As a result, these methods are still limited to non-shared policy generation rather than exploring multi-agent shared policy and cooperation under mixed autonomy.

Real-world automated driving problems often involve multiple agents in a dynamic and partially observed environment. An emerging question is how to promote cooperation among agents instead of relying on ego/selfish-driving. Shalev et al. (2016) introduce a hierarchical temporal abstraction with a gating mechanism that significantly reduces the variance of the gradient estimation in multi-agent automated driving environment. Furthermore, Palanisamy et al. (2019) use Partially Observable Markov Games (POSG) to formulate the connected automated driving problems. Their approach can be trained in both centralized and decentralized frameworks. Wang et al. (2019) develop the cooperative lane change system by considering the overall traffic efficiency instead of the travel efficiency of an individual vehicle, which can lead to a more harmonic and efficient traffic system rather than competition.

However, it remains unknown how to better utilize information of surrounding agents to encourage cooperation and make the driving behavior more efficient. Recent research progress on graph information sharing has brought new and promising perspectives to the multi-agent reinforcement learning problems (Wu et al., 2020). Iqbal et al. (2019) propose to use a multi-head attention mechanism to enable effective and scalable learning in complex multi-agent environments. However, this framework doesn't consider training model's parameter sharing among neighbors, which may make it hard to train and implement into the CAV setting. Agarwal et al. (2019) propose to create a shared agent-entity graph and introduced curriculum learning to increase transferability. Jiang et al. (2018) propose the graph convolutional reinforcement learning approach for multi-agent to learn cooperative strategies. However, for a highly dynamic environment, such as the automated driving setting, not only vehicles in close-range but also vehicles with high relative speed to the ego vehicle should be considered. Previous studies have not fully utilized both position and speed information from surrounding agents, which hinders the feasibility of real-world implementation of CAV.

2.3. Cooperation in multi-agent system

In this thesis (dealing with mixed autonomy system research), the system-level safety and efficiency are focused on system-level improvement. In other words, each agent needs to learn to

cooperate with others instead of just focusing on improving their self-interest. The essential way to encourage cooperation is to learn interactions among each agent. Park et al. (2019) propose concatenating each agents' feature then average the combined feature into the interaction network to aggregate multi-agent information. Mean-field control utilizes a central controller that coordinates all agents' behaviors. Yang et al. (2018) propose the mean-field reinforcement learning in which they approximate the interactions within the population of agents using the average effect from the overall population or neighboring agent. This mechanism can enforce the interplay between two entities in order to encourage cooperation. Bacchiani et al. (2019) extend the asynchronous advantage actor-critic approach in a multi-agent scenario, allowing every agent to learn to interact with other similar agents. However, the aforementioned literature needs the central network to integrate all the information together. In our setting, each agent can only observe local information. As a result, this thesis develops the decentralized control framework with parameter sharing for multi-agent control.

2.4. Traffic signal control

In Chapter 4, this thesis is focusing on intelligent traffic signal control. In the following section, typical traffic signal control solutions (methods) will be illustrated, and the way to improve robustness in decision-making will also be analyzed.

Conventional coordinated methods. These methods usually coordinate traffic signal control by modifying the time interval between each traffic signal phase. The limitation of these methods is that they can only optimize the traffic flow in the fixed directions. It's difficult to be applied in large-scale networks or irregular road networks. Several advanced methods are the fixed time method and MaxPressure method. The fixed time method uses a pre-determined plan for cycle length and phase time, which is widely used when the traffic flow is steady (Koonce et al., 2008). MaxPressure is a popular and strong baseline for network-level traffic signal control methods in the transportation area. At each time step, it selects the action that maximizes the number of moving vehicles from inbound lanes (varaiya et al., 2013). However, it's very likely to get into local optima.

RL-based Traffic Signal Control. The first implementation of RL in TSC uses tabular Q-Learning to learn from a single intersection (Wiering et al., 2004, Cai et al., 2009) then used RL with function approximations. However, most previous investigations are limited to toy scenarios. To develop RL method for more realistic traffic data, researchers turned their attention to deep RL. Wei et al. (2018) show that deep reinforcement learning can dynamically adjust to real-time traffic. However, the high dimension of the joint action space still limits the scalability of centralized RL approaches.

Large-Scale Traffic Signal Control. Multi-agent Reinforcement Learning (MARL) is introduced to improve the scalability of RL agents by using a decentralized control framework. Chu et al. (2019) use advantage actor-critic (A2C) as a large-scale TSC method. To be specific, neighbors' information is adapted to improve sample efficiency and promote cooperative strategy; further, a spatial discount factor is introduced to improve the learning efficiency, i.e., reduce fitting difficulty. To enable cooperation of traffic signals, recent works study how to encourage cooperation through graph representation learning. Wei et al. (2019) propose to use a graph attention neural network in the setting of large-scale road networks with hundreds of traffic signals. They model each TSC as an agent. Agents learn to communicate by attending to the representations of neighboring intersections. Their results demonstrate the effectiveness of the attention mechanism to help cooperation and achieve superior performance over state-of-the-art methods. Recently, Devailly et al. (2020) further exploit the vehicular data at its finest granularity by representing every vehicle as a node. They demonstrate the flexibility of GCNs, which can enable transfer-ability to unseen road networks. However, neither works evaluate their methods under exogenous uncertainties.

2.5. Robustness of decision making system

As stated in Chapter 4, a robust decision-making model is needed to solve exogenous uncertainty problems. In transportation research, a very straightforward way to solve the exogenous uncertainty problem from sensor failure is to use imputation methods. For example, recent work uses a variational Bayes approach to predict missing values accurately (Chen et al., 2019). Graph Neural Network (GNN) can also be an efficient and effective tool for recovering information from

malfunctioned sensors (Wu et al. 2020). Similar methods have also been seen in the reinforcement learning community. For example, Mai et al. (2019) formulate the problem as inverse reinforcement learning to recover the expert's reward function and calculate the likelihood of the demonstrated trajectories given missing observation pairs. Bayesian multiple imputation and bootstrap have also been used to approximate the distribution of the training set in order to estimate the state-action value function given missing data (Lizotte et al., 2008).

Such methods have not been adapted for TSC and, in any case, are not tailored to the problem of demand surge. Recently, deep RL has proved to be robust under the impact of special events, such as demand surges, sensor failures, and partial detection. Rodrigues et al. (2019) develop the callback-based framework to enable flexible evaluation of different deep RL configurations under special events. They conclude that when training in scenarios with sensor failures, the RL approach can be quite robust to the widely sensor failure and demand surge problems. Zhang et al. (2020) demonstrate that deep RL agents can be robust within the partially detected intelligent transportation systems (PDITS), which is a partially observable Markov decision process (POMDP) in the RL community, in which only part of vehicle information can be acquired. They have conducted experiments under different detection rates and report that RL based control method can improve travel efficiency even with a low detection rate. However, their evaluation scenario is limited to 1 to 5 intersection cases. Most importantly, they only empirically demonstrate the robustness of the existing deep reinforcement learning approach but have not further discussed how to improve the robustness based on previous reinforcement learning methods.

Chapter 3. Efficient Connected and Automated Driving

3.1. Mixed autonomy control problem formulation

In this thesis, the mixed autonomy control problem consists of a mixed of connected automated vehicles and human driving vehicles (the black vehicle stands for human-driving vehicle and the blue vehicle stands for automated vehicle). Road sections with and without traffic controls (interrupted and uninterrupted traffic conditions) are considered. This problem is formulated as below:

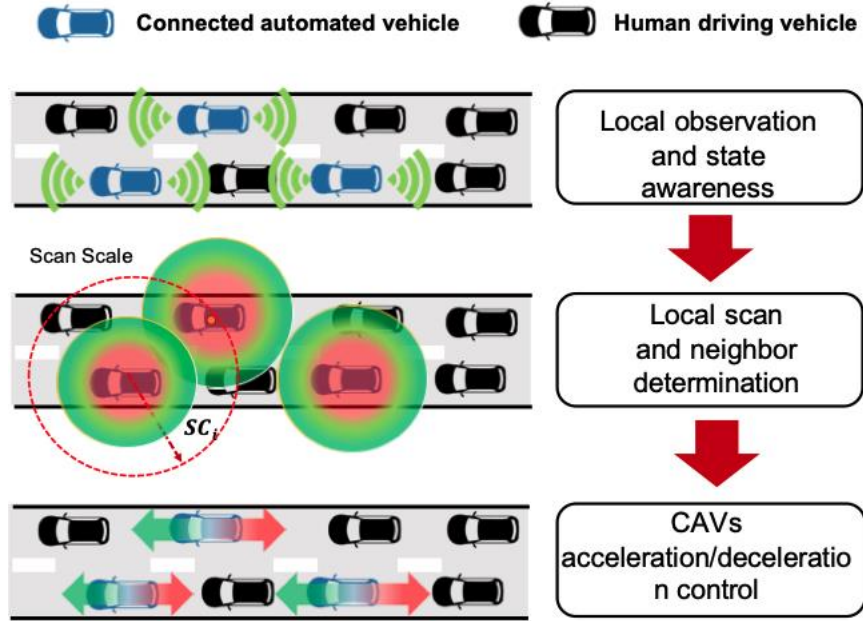


Figure 3.1: CAV control framework

Blue vehicles represent the connected and automated vehicles (CAVs), while the black vehicles represent the human-driving vehicles. The colormaps stand for the Gaussian speed field (Zhang et al., 2021) of each CAV (as shown in the colormap). The red dotted line stands for the scan scale, which is the local observation range of each agent (scanning sensing area).

Following the work of Kreidieh et al., 2018, N CAVs is modeled as N homogeneous agents in a mixed-autonomy traffic network to achieve better generalization ability. Their decision procedures can be divided into three stages: (1) at the beginning of each decision, the agents c_i ,

$i=1, \dots, N$, will first have a local observation and identify their current state s_t^i ; (2) then each agent will manage to locate and communicate with their neighbors; (3) once the agents acquire both information from themselves and neighbors, they will make decisions to accelerate/decelerate accordingly.

Formally, the task in such a mixed-autonomy transportation system can be defined in the setting of multi-agent reinforcement learning (MARL) with the following components:

1) Agent design

In the simulation, two types of agents are considered: human-driving agents whose acceleration or deceleration decisions are determined based on car-following models (e.g., intelligent driver model (bando et al., 1995)); CAV agents which are controlled by deep reinforcement learning framework.

2) State observation

State observation is defined as $o_i, i=1, \dots, N$, for each CAVs, which consists of speed and position of the ego vehicle, as well as the relative speed and position from the ego vehicle, its leader, and follower. The state observation from for CAV i is denoted by $S_i = \{o_i^m, \dots, o_i^m\}$.

3) Action space

Action $a_i, i=1, \dots, N$ is the speed adjustment for each CAVs, bounded by the maximum acceleration and deceleration specified in the environment's parameters.

4) Reward function

Reward functions which are defined differently for different simulation scenarios. $r_i, i=1, \dots, N$ is used to denote the reward for each CAVs.

For the ring scenario and the figure-eight scenarios, the reward function is defined to encourage high average speeds from all vehicles in the network and to penalize accelerations/decelerations by the CAVs.

The reward for *ring* and *eight* scenarios is defined as (Wu et al., 2017):

$$r = -w_v * (\bar{v}_t - \bar{v}) + w_a(\bar{a}_{the} - \bar{a}), \quad (3.1)$$

where w_v and w_a stand for the weight parameters for average velocity and average acceleration. \bar{a}_{the} is the threshold of the acceleration.

For the merge scenario, reward function encourages similarity of the system-level speed to a desired speed, while slightly penalizing short headways among CAVs (Wu et al., 2017).

$$r = -w_v * (\bar{v}_t - \bar{v}) + w_h * \left(\min \left(\frac{(\bar{h} - t_{min})}{t_{min}}, 0 \right) \right), \quad (3.2)$$

where w_v and w_h stand for the weight parameters for average velocity and average headway. \bar{v} is the average velocity, \bar{v}_t is the target velocity, \bar{h} is the average headway, t_{min} is the smallest acceptable time headway, which is defined as 1 s.

5) Termination

An episode is terminated if the time horizon is reached or a collision happens.

3.2. Methodology

3.2.1. Multi-agent cooperation within mixed-autonomy system

In previous literature (e.g., Shi et al., 2019, Wang et al., 2018), CAV was designed to have an individual policy under the environment. This is not applicable for controlling a group of CAVs to learn in a mixed-autonomy transportation environment due to training complexity. Therefore, the shared policy is introduced into the proposed control framework.

Based on the graph attention on CAVs, the multi-agent reinforcement learning architecture is established. 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 (Sutton et al., 2018), as shown below:

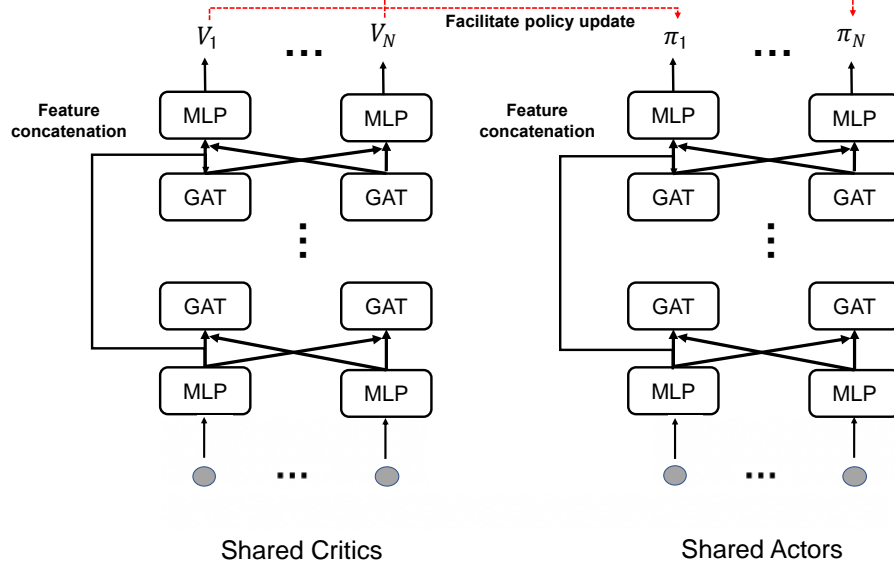


Figure 3.2: Architecture of MARL-CAVG

The architecture of MARL-CAVG is shown above in Figure 3.2. The critic network is a graph convolutional neural network, i.e., GAT, 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_t). It will be further used for advantage estimation to train the actor network.

In this thesis, several techniques are designed and integrated 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 Figure 3.1, after observing the 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., Jiang et al., 2018, Wei et al., 2019), the adjacency matrix is built based on the Gaussian speed field using the Gaussian process regression (GPR) model (Zhang et al., 2020). The standard exponential kernel function is computed as:

$$K(x_i, x_j) = A * \exp\left(-\frac{(x_i - x_j)^2}{2\sigma^2}\right), \quad (3.3)$$

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 impose less correlation on two points that are far away. In our research, length scale is fixed as 4m (Zhang et al., 2020).

Furthermore, the feature representation can be dynamically constructed in the adjacency matrix at time step t , whose elements are defined as follows:

$$M_t(i, j) = K(x_i, x_j) \Delta V(x_i, x_j), \text{dis}(x_i, x_j) \leq SC_i. \quad (3.4)$$

The location information is incorporated (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. Notably, each row i represents an ego vehicle, and each non-zero element of this row is the neighboring information between this ego vehicle and surrounding vehicles within its scan scale SC_i . Intuitively, ego vehicle will be more sensitive to closer surrounding vehicle than a more distant one.

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

$$h_i^k = f(\text{concat}[M_t H^{k-1}, D_i^{-1} M_t H^{k-1}] W_i), \quad (3.5)$$

where f is the activation function 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 $H^{k-1} = [h_1^{k-1}, \dots, h_N^{k-1}]$.

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 N_i neighbors of ego CAV i , the attention score on neighboring CAV j can be computed as:

$$q_i = f^{query}(h_i * W^{query}), \quad (3.6)$$

$$k_j = f^{key}(h_j * W^{key}), j \in N_i \quad (3.7)$$

$$\phi_{i,j} = softmax(\frac{q_i * k_j^T}{\sum_{l \in N_i} q_i * k_l^T}), j \in N_i. \quad (3.8)$$

Note that the layer index is omitted here for simplicity. We use f^{query} and f^{key} to encode input features as query-key pairs, then dot-product between query q_i and key k_j vectors is conducted. With *softmax* activation function it can further quantify the strength of relationship $\phi_{i,j}$ between two entities (Vaswani et al., 2017). With the attention scheme, an ego CAV 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 an 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 (Silver et al., 2017)), the CAVs need to generate continuous action space for smooth and efficient control strategy.

Therefore, Proximal Policy Optimization (PPO) is used (Schulman et al., 2017) for CAVs to handle continuous action space. The critic network is designed as a graph convolutional neural network parametrized by ϕ . Notably, the output of critic network at each time step t is state value estimation V_t (i.e., short for $V(S_t, M_t)$), it will be further used for advantage estimation to train the actor network.

The update of the gradient for critic is based on temporal difference learning (Sutton et al., 2018):

$$\nabla_{\phi} L(\phi) = \nabla_{\phi} E[\sum_{n=1}^N (r_i^n + \hat{V}(S_{t+1}, M_t) - \hat{V}(S_t, M_t))^2], \quad (3.9)$$

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 $A_i^t, i = 1, \dots, N$ from critic:

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} E_{\pi_{\theta old}} [\sum_{i=1}^N \min(r_i^t(\theta) \hat{A}_i^t, \text{clip}(r_i^t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i^t)], \quad (3.10)$$

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 (e.g., simulation resolution is less than 1 s). In addition, on-policy roll-out is performed to collect the experience (i.e., $\{o_i, a_i, r_i\}$) and the advantage estimation for agent i at step T is calculated as: $\bar{A}_i^T = \sum_t^T \gamma^t r_i^t - \hat{V}(S_t, M_t)$.

The overall training algorithm is summarized in Table 3.1:

Table 3.1: Training algorithm for CAV control

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 $B = \emptyset$, batch size as b .
Initialize parameters ϕ, θ for critic and actor network.
for each episode do
for $t = 1$ to T do
Obtain state observation $o_i, i = 1, \dots, N$ and global observation $S = [o_1, \dots, o_N]$.
for CAV $i = 1, \dots, N$ do
Sample action a_i from $\pi^{old}(S, M_i \theta)$ to control CAV.

```

end for

Obtain next state observation  $(o'_i)_{i=1}^N$  and global observation  $S' = [o'_1, \dots, o'_N]$  as well as the reward signal  $r_i$  for each CAV.

 $B \leftarrow B \cup (a_i, o_i, o'_i, r_i, M_i)_{i=1}^N$ 

if  $|B| \% b = 0$  then

    Fetch experience from  $M$  and perform roll-out

    Update  $\phi$  based on Equation 3.9

    Update  $\theta$  based on Equation 3.10

     $B = 0$ 

end if

if collision happened then

    Break

end if

end for

end for=0

```

3.3. Experiment analysis

This section provides analysis on the experiment settings and algorithms' setup. The models' performance is also compared with several baselines.

3.3.1. Experiment setup

Extensive experiments are conducted in Flow, an open-source project that supports mixed-autonomy control. The proposed algorithm is evaluated based on several benchmarks, car-following (Kreidieh et al., 2018), intersection and merge (Vinitsky et al., 2018) as shown in which

are common intelligent traffic control scenarios. In the simulation, the horizon represents the number of steps per roll-outs, and each time step is 0.1s.

1) Car following control

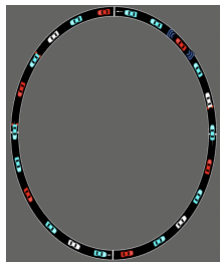
This is a common scenario in the highway without bottleneck. To simplify the training process, this thesis considers a ring-shaped network with a single lane, which is shown in Figure 3.4-(a). In the initial condition, all the vehicles are uniformly distributed on the circular road with the same initial speed. Experimental results in Sugiyama et al. (2008) 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.

2) Intersection control

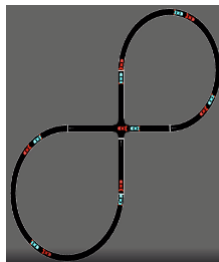
This is a common urban traffic scenario. As shown in Figure 3.4-(b), in this case, CAV control can help improve the overall travel efficiency of urban transportation systems. A simple intersection is considered with a figure-eight shape network with one or two circular tracks.

3) Merge

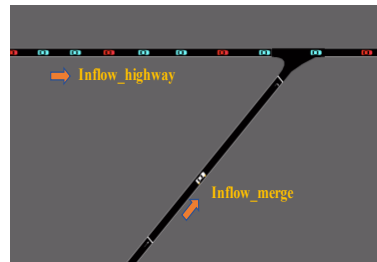
This scenario is common in highway networks. Vehicles move from the on-ramp create backward propagating stop-and-go waves. As a result, perturbations will propagate upstream from the merge point and reduce the throughput of vehicles in the network. For the merge scenario, the total number of vehicles is considered as the number of vehicles per hour coming into the highway lane. See Figure 3.4-(c) for the implemented merge scenario.





(a) Ring network



(b) Figure eight network



(c) Merge network

 Automated vehicle agent
 Observed humans



 Unobserved humans
 Sensing of leader and follower vehicle

Figure 3.3: Road network structures

Different road networks are given above. To be specific, several variants are considered in the simulation scenario to test the performance of the proposed model. Firstly, for ring networks, it is a standard car-following evaluation scenario that is common in the real-world. Secondly, for the figure-eight network, it's a more challenging scenario compared to a ring network with the 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 to the dynamic changing environment.

3.3.2. Algorithm setup

The proposed MARL-CAVG method is compared with several state-of-the-art baselines, including not only reinforcement learning frameworks (single-agent and multi-agent) but also car-following models in traffic flow theory. For all experiments, we run 100 episodes with a collection of the average results of 10 random seeds which is similar to the setting in this study (Wu et al., 2017). The explanations for selecting these baselines are given as follows.

1) Intelligent driver model (IDM):

Intelligent driver model (IDM) (Bando et al., 1995) 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., (Wang et al., 2018, Shi et al., 2019)). A 0.2 random noise is added to the action to model the uncertainty of human-driving behavior.

2) Deep Deterministic Policy Gradient (DDPG):

Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015): DDPG is a deterministic version of a model-free RL algorithm to deal with continuous action space. CAV agents can reliably learn the optimal policy with continuous actions. A single-agent is constructed for the training framework based on the DDPG method, which is similar to Huang et al., 2019.

3) Proximal Policy Optimization (PPO):

Proximal Policy Optimization (PPO) (Schulman et al., 2017): PPO is a gradient-based RL algorithm to deal with continuous action space. Unlike DDPG, PPO is an on-policy algorithm.

4) Multi-agent Deep Deterministic Policy Gradient (MADDPG):

Multi-agent Deep Deterministic Policy Gradient (MADDPG) (Lowe et al., 2017): This is a widely used multi-agent framework with centralized critics and decentralized actors. This is a baseline model without introducing a graph neural network to consider information from neighbors specifically.

5) 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.

3.3.3. Performance comparison

Several experiments are conducted in these three networks.

1) Evaluation in car-following control:

Figure 3.5 shows the training performance of different methods in the car-following control scenario. As can be seen, the MARL-CAVG method outperforms other methods with a large margin. From Table 3.1, it can be seen that MARL-CAVG achieves the second-highest velocity and the smallest acceleration, which makes it achieve the highest return in this scenario. It indicates that the proposed model can better mitigate the shock-wave during the car following control.

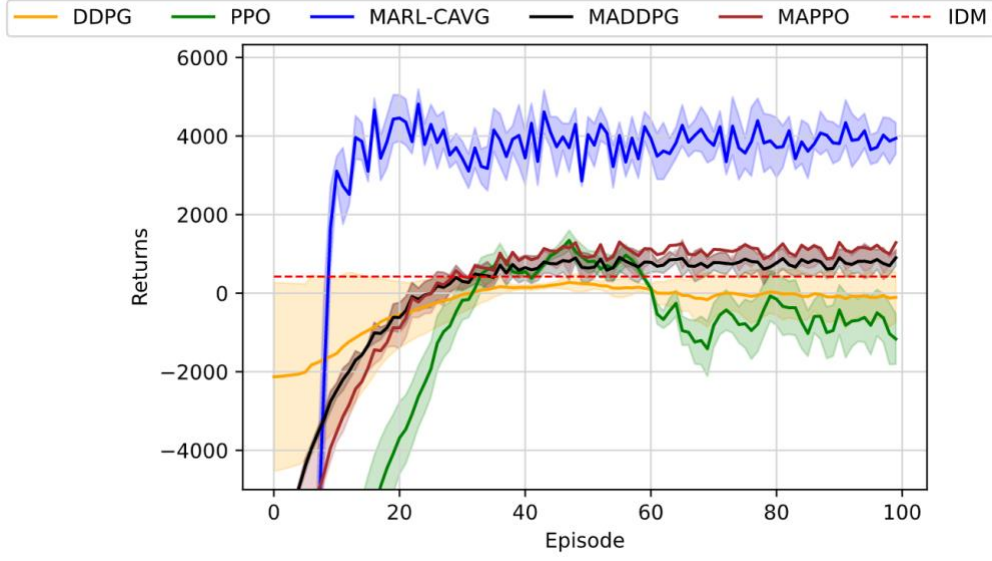


Figure 3.4: Learning curve in ring network

In the ring network, 3000 horizons are used, the total vehicles are 22, and autonomous vehicles are 16. The lane is 1 and target speed is 30km/h.

Table 3.2: Performance comparison in ring network

Methods	Velocity(m/s)	Acc ($0.1 \text{ m}^2 / \text{s}$)	Return
IDM	2.754	3.318	424.12
DDPG	3.134(± 0.148)	2.718(± 0.378)	70.063(± 70.3506)
PPO	3.165(± 0.145)	3.129(± 0.369)	-676.959(± 65.046)
MADDPG	3.270(± 0.148)	2.121(± 0.366)	779.140(± 29.178)
MAPPO	3.379(± 0.142)	2.782(± 0.325)	776.225(± 20.121)

MARL-CAVG	3.391(± 0.092)	0.835(± 0.171)	2593.99(± 51.820)
-----------	----------------------	----------------------	-------------------------

2) Evaluation in intersection control:

In the figure-eight scenario, the concern 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 Figure 3.6, it can be found that MARL-CAVG method can achieve better performance when considers the aforementioned changes. It can achieve better control performance, maintaining a good balance between safety and efficiency. From Table 3.2, it can be found 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 deal with the trade-off, which is beneficial to get the highest cumulative return.

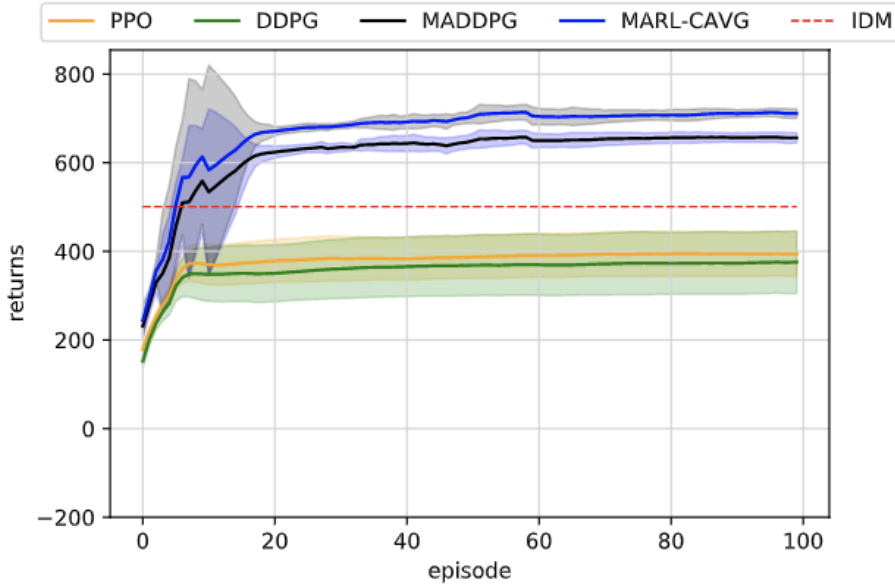


Figure 3.5: Learning curves in figure-eight network

In figure-eight network, 1500 horizon is used, the total vehicles are 14 and the autonomous vehicles are 7 and target speed is 30km/h.

Table 3.3: 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(± 2.091)	8.619(± 3.191)	379.061(± 46.852)
PPO	4.0654(± 2.415)	8.812(± 4.090)	-357.946(± 64.509)
MADDPG	4.879(± 1.231)	6.192(± 2.213)	618.641(± 32.796)
MARL-CAVG	4.265(± 1.913)	3.123(± 1.139)	669.119(± 28.895)

3) Evaluation in intersection 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, it can be found that our model outperforms the baselines in most evaluation indicators.

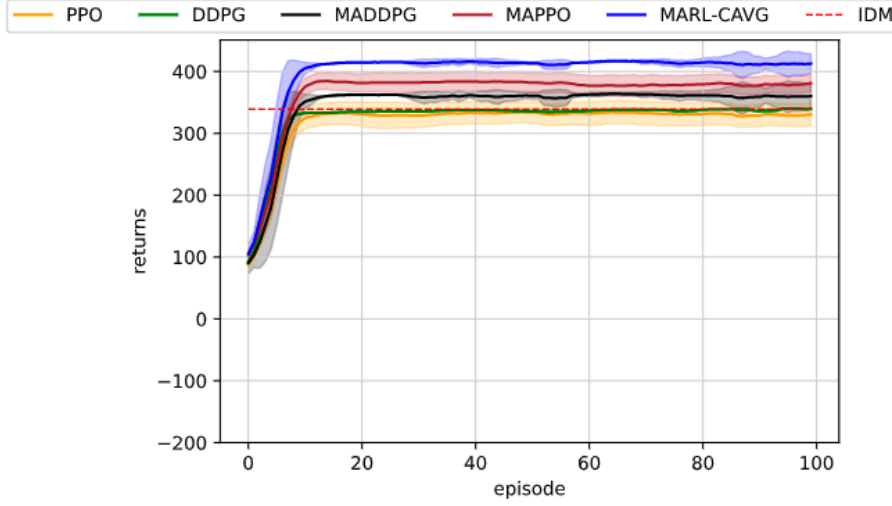


Figure 3.6: Learning curves in merge network

In the experiment, 600 horizons are used, 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.

3.3.4. Further analysis

1) Visualization of Control Performance

To evaluate the control performance, the ring network is selected as an example and plot space-time diagram and velocity figures with the trained policy after 200 episodes. The number of heads in the attention module is 8. Each method is tested 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.

As shown in the result of Figure 3.8, the red curve stands for the control performance with all human driving vehicles, which is unstable. After automation is turned on, the flow becomes stable. It can be seen that after automation turns on, the velocity will become stable. Furthermore, the proposed model can reach the highest speed compared to other baselines.

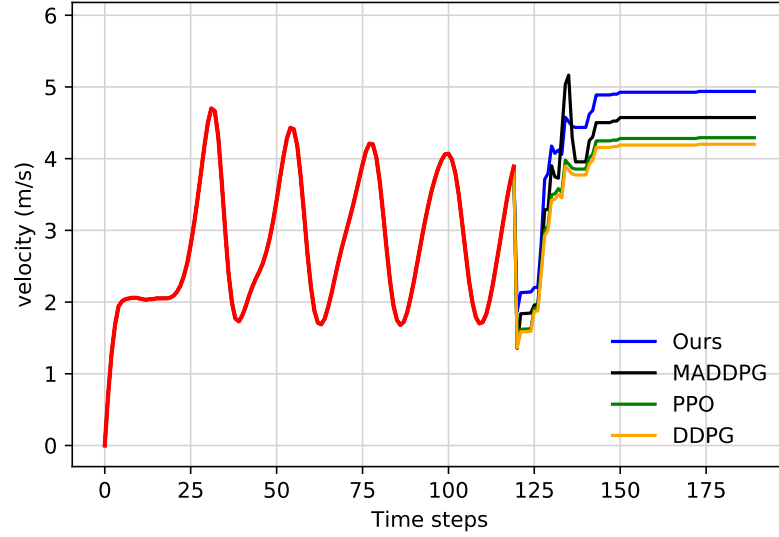


Figure 3.7: Velocity performance in ring network

To visualize the impact of shock-wave, the space-time diagram performance in the ring network before and after the automation is turned on is further compared. It can be seen from Figure 3.9 that the velocity fluctuates sharply. With automation turned on, the velocity becomes smooth, and the average velocity increases, as shown in Figure 3.10.

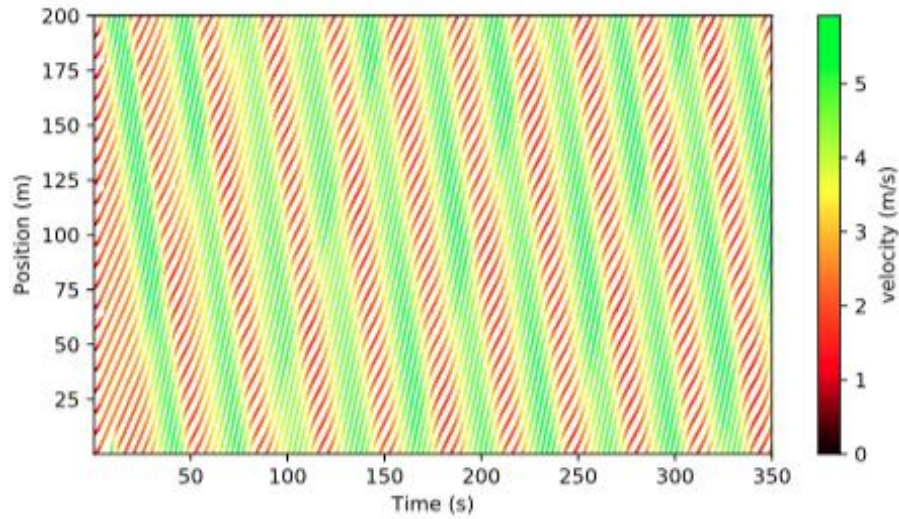


Figure 3.8: Space time diagram without automation

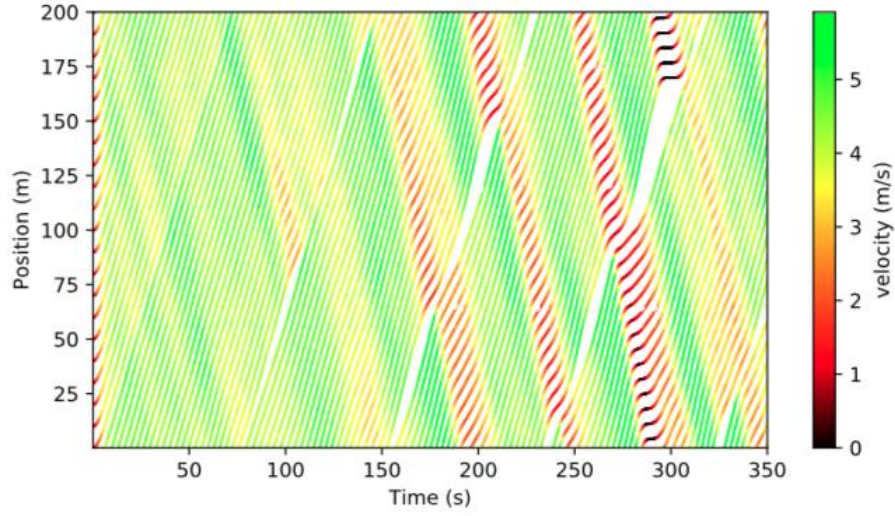


Figure 3.9: Space time diagram with automation

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 the proposed model under different penetration rates, several typical ring scenarios are selected to make the comparison. The typical multi-agent RL approach (MADDPG) and single-agent RL approach (DDPG) are selected as the baselines. As shown in Figure 3.11, it can be seen 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 autonomous 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 increasing number of controlled agents, and therefore achieve increasing return and the best overall performance.

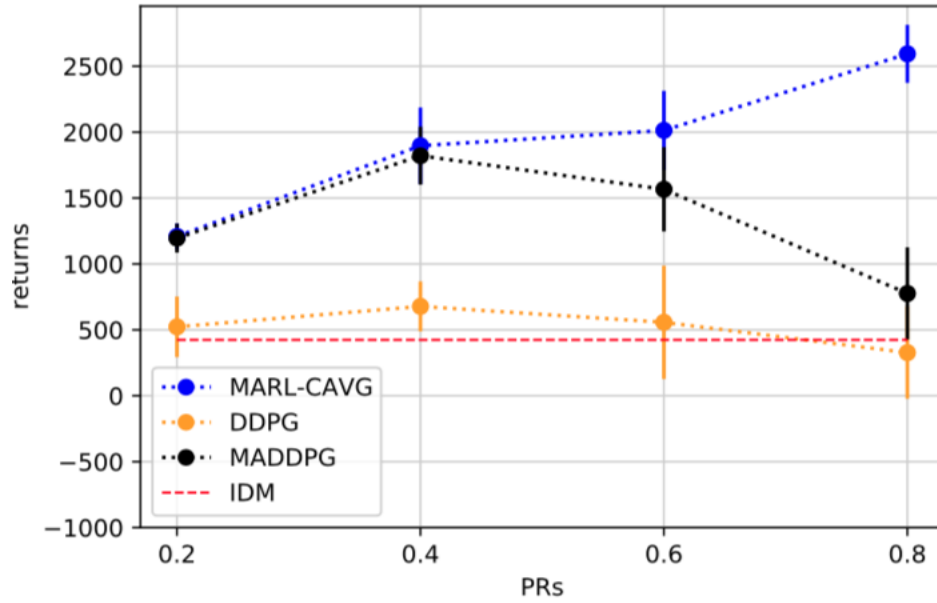


Figure 3.10: Evaluation of different penetration rates

3) Evaluation of different target speeds

Different target speeds of each vehicle will reflect different driving behaviors (e.g., a more aggressive driver tends to set a higher target speed in his trip). A higher target speed will increase the travel efficiency but tend to have safety problems because of large acceleration/deceleration. In this subsection, the penetration rate is fixed as 0.4, and then test with different target speeds, then evaluate different methods' performance. The target speed is set as 20km/h as the baseline, then calculate the percentage (%) of increase for each method based on their 20km/h baseline. As shown in Figure 3.12, it can be found that for each method, with the increase of target velocity, the agent's performance will be better. However, when the target velocity is too high (e.g., 120km/h), then the agent's performance will decrease. The proposed model achieves the best return given the highest target velocity. This demonstrates that the proposed model is more robust to different driving behaviors.

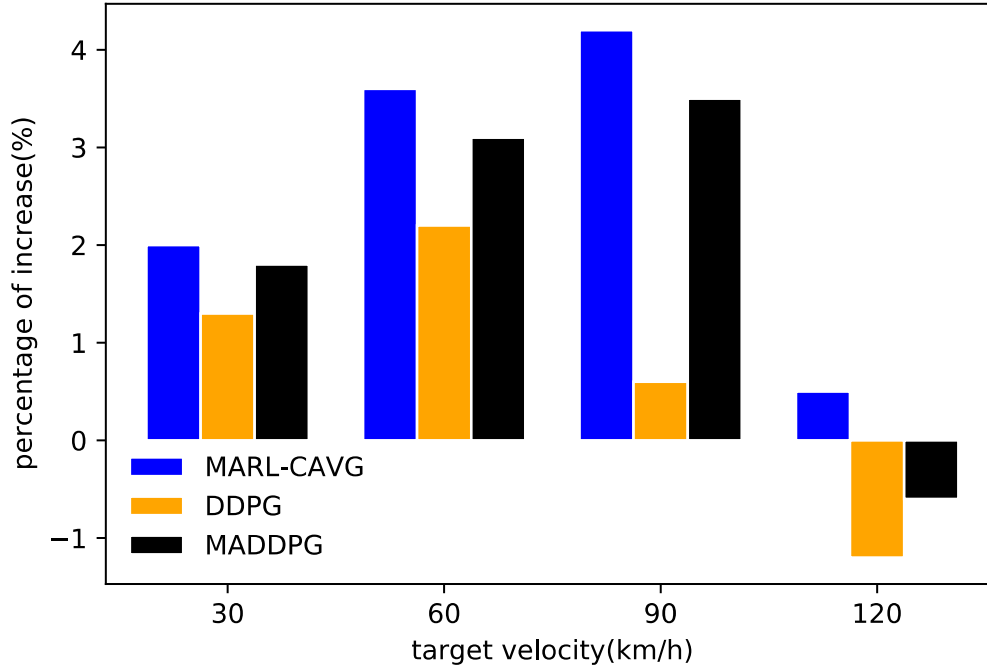


Figure 3.11: Evaluation of increase of returns under different target speeds

4) Evaluation of different architectures

For different architectures of the model, the range of scan scales on model performance is evaluated. The results are shown in Figure 3.13. From the results, it can be found that if slightly enlarge the scan scale, the performance will also increase 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.

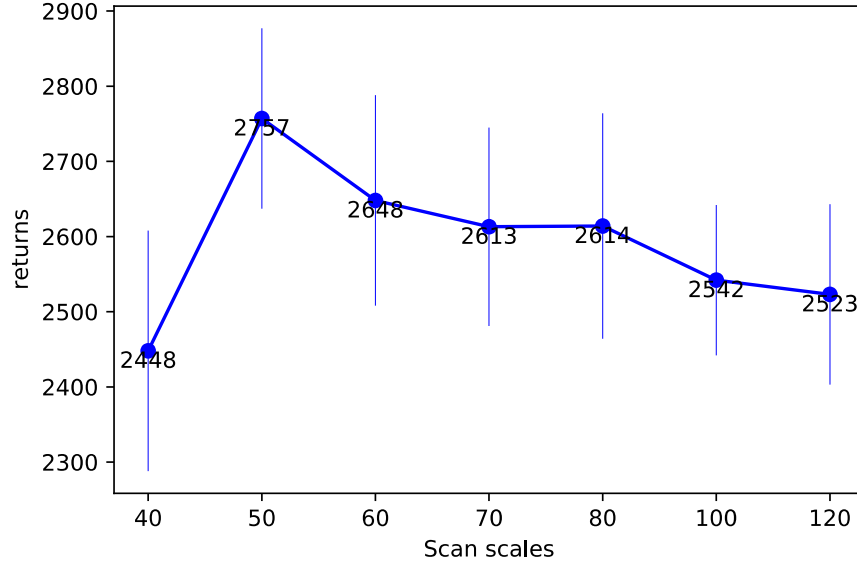


Figure 3.12: Evaluation of different scan scales

Furthermore, the performance in terms of different information used to build the adjacency matrix is also evaluated. Only speed information or position information and both speed and position information are considered as given in Equations 3.11 and 3.12. The different information considered is as follows:

Only consider position information:

$$m_i = x_i - x_o, (3.11)$$

where x_o and x_i are the position of the ego agent and surrounding i^{th} agent.

Only consider velocity information:

$$m_i = \frac{v_t}{v_o (|v_i - v_o| + \epsilon)}, (3.12)$$

where v_o and v_i are the velocity of the ego agent and surrounding i^{th} agent.

Intuitively, 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 more considerable. 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 safe or efficiency consideration. Therefore, the measure will be smaller, and priority will be higher.

Table 3.4: Returns with different information in adjacency matrix

Adjacency matrix	Position	Velocity	Both
Returns	2490.99 (± 20.149)	2601.87(± 19.825)	2710.32(± 23.581)

The penetration rate is fixed as 0.4, target speed as 30km/h. As shown in Table 3.4, 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, experiments with/without (i.e., head=0) the attention module and the different number of heads in the attention module are conducted. In the experiment, the penetration rate is 0.4. The target speed is 30km/h. Only a different attention module has experimented with.

Table 3.5: Returns with different heads in attention module

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

From Table 3.5, it can be found that without attention module (head=0), the performance of the model decreases a lot. The increase of attention heads will increase the model's performance, while too many heads (heads \geq 10) will decrease the performance.

3.4. Main findings

In this chapter 3, the graph convolutional reinforcement learning approach for CAV control by encouraging efficient cooperative traffic control in mixed autonomy is proposed. There are several interesting findings. Firstly, the shared policy and efficient communication strategy, i.e., graph attention, can help agents efficiently cooperate with each other. The proposed model can achieve the best performance in all scenarios. It can learn to balance safety and efficiency. Secondly, the proposed model also demonstrates robustness to different penetration rates, target speeds, and network structures. However, in this chapter, sensor failures and demand surges have not been considered. In the next chapter 4, how to improve robustness to these exogenous uncertainties will be further discussed.

Chapter 4. Robust Large-scale Traffic Signal Control

4.1. Traffic signal control problem formulation

In this section, the traffic signal control problem is formulated as below:

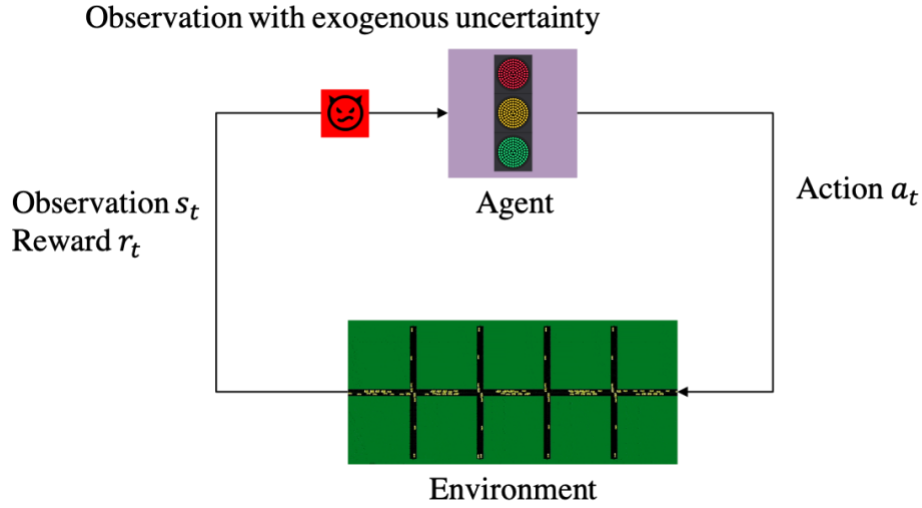


Figure 4.1: Reinforcement learning for TSC with exogenous uncertainty

1) Agent design

Each TSC is the agent in this problem, which can only observe the local information of the surrounding environment. Hence, the problem is a partially-observable MDP.

2) State observation

TSC node: it represents the state of a controller. The features include the number of seconds since a traffic controller performed its last phase switch.

Lane node: it represents the state of a lane. It includes the length of the lane.

Connection node: it represents the state of an existing link between an entry lane and an exit lane. For example, the connection exists between an entry lane A and an exit lane B if a vehicle

on A is allowed to go to continue its travel to B. The features in the connection node are whether a connection is opened under the current phase; whether if an open connection between an entry and an exit lane it has priority or not; the number of switches the controller has to perform before the next opening of a given connection; and whether the next opening of the connection will have priority or not.

Vehicle node: it represents the state of a vehicle which includes its current speed and position on the current lane as a feature.

3) Action space

At every intersection of the road network, a predefined logical program, composed of a given number of phases, depending on the roads, lanes, and the connection information. The program is given by the road network. The agent's action is to choose whether to switch to next phase or prolong the current phase, as a result, it's a binary action.

4) Reward function

Each agent i can obtain a reward r_i at time t from the environment. In this thesis, the goal is to maximize the travel efficiency of the vehicles by reducing queue length. The reward is defined as the negative sum of total queues lengths per intersection q , $r_i^t = -\sum_l q_{i,l}^t$. where $q_{i,l}^t$ is the queue length on the lane l at time step t .

4.2. Methodology

4.2.1. Overview of learning process

In this framework, as shown in Figure 4.2, each vehicle (V), lane (L), connection node (C), and traffic signal controller (TSC) are abstracted as nodes in the graph. The information of each node and its connection can be exploited through graph representation learning using a GCN. At the output of the GCN, it can be obtained that a graph representation embedding ψ . These can be trained using an RL objective or a DRL objective. In DRL, those features are combined with an

embedding function $\phi(\tau)$ where τ is the quantile, using the dot product. Results demonstrate that combining the DRL and the standard RL objectives improves performance.

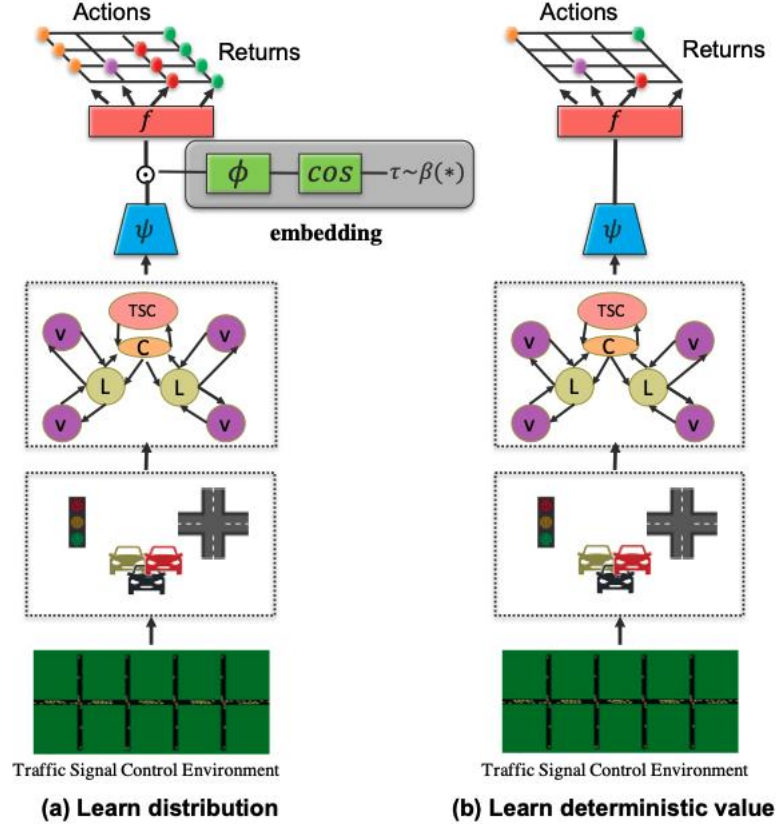


Figure 4.2: DGRL framework overview

As shown in Figure 4.2, from (a), learn the distribution of returns while from (b) learn deterministic value.

4.2.2. Graph representation learning on different nodes

1) Graph Representation Learning on Different Nodes

The traffic-signal control system involves a traffic signal controller, lanes, connections between lanes, and vehicles (e.g., nearby ones). In this thesis, TSC nodes, connection nodes, lane nodes, vehicle nodes are modeled as the entities within the proposed GCN structure like in (Devailly et al., 2020).

Every layer of the GCN uses one set of parameters per edge type to perform message-propagation:

$$H^{(l+1)} = \sigma \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^l W^l \right), (4.1)$$

where $H^{(l+1)}$ denotes the extracted features from $(l+1)$ layer, A is the adjacency matrix, D is the degree matrix, W^l is the weight matrix which contains the parameters in l embedding layer and σ is the activation function.

Consider the GCN network in above Equation. Let $\psi: X \rightarrow R^d$ be an embedding function parametrized by the GCN layers. Then, add a subsequent fully-connected layer to map $\psi(x)$ to the estimated action-values, such that $Q(x, a) \equiv f(\psi(x))_a$ where a in $f(\cdot)_a$ indexes the output action.

Then the predicted Q values can be derived (Mnih et al., 2015):

$$\hat{Q} = H^L W_p + b_p, (4.2)$$

where $W_p \in R^{c \times p}$ and $b_p \in R^p$ are parameters of the neural networks, p is the number of phases (action space). L is the number of GCN layers. Here, H^L can be considered as same as ψ in Figure 4.2.

Then the loss function of learning deterministic values can be represented as:

$$L_{MARL} = \frac{1}{N} \sum_{j=1}^N (y - Q(s_t, a_t))^2, (4.3)$$

where $y = r_t + \gamma \max_a Q(s_{t+1}, a_{t+1})$, N is the number of intersections in the whole road network. θ represents trainable parameters.

2) Parameter sharing

To enable the transfer ability and training on a variety of networks / architectures, parameter sharing is considered for all decision processes, including inside and outside of a given

decision process (e.g., between two same type nodes on the same intersection and two same type nodes on unrelated intersections).

4.2.3. Decentralized distributional RL for TSC

The previous section introduces the GCN and a standard RL objective. Now, this section will discuss learning the GCN model using distributional RL (DRL). Compared to traditional RL, DRL models the distribution over returns. The expectation of that distribution yields the standard value function. In this thesis, implicit quantile networks (Dabney et al., 2018) are used as a distributional version of Deep Q-Networks (Silver et al., 2017). Implicit quantile networks can approximate any distribution over returns and show superior performance compared to other DRL methods.

Implicit quantile networks define an implicit distribution using samples τ from a base distribution $\tau \sim U([0,1])$. The implicit distribution is parametrized using $\phi: [0,1] \rightarrow R^d$. The function ϕ provides the embedding for quantile τ . This embedding ϕ is combined with the GCN's output embedding ψ to form the approximation of the distributional Q-values (see Figure 4.2 -(a)):

$$Z_\tau(x, a) \equiv f(\psi(x) \odot \phi(\tau))_a, \quad (4.4)$$

where \odot represents the element wise product, the a on the RHS indexes the output of the function f . As in the original IQN paper (Dabney et al., 2018):

$$\phi_j(\tau) := ReLU\left(\sum_{i=0}^{n-1} \cos(\pi i \tau) w_{ij} + b_j\right), \quad (4.5)$$

where n , a hyperparameter, is the size of the input embedding, $j \in 1, \dots, d$ indexes different units (neurons), and w_{ij} and b_j are parameters shared across all TSCs (much like parameters of the GCN are also shared across TSCs).

As a result, the state-action value function can be represented as the expectation:

$$Q(x, a) := E_{\tau \sim U([0,1])}[Z_\tau(x, a)]. \quad (4.6)$$

Its associated greedy policy is:

$$\pi(x) = \operatorname{argmax}_{a \in A} Q(x, a). \quad (4.7)$$

As described in the IQN paper (Dabney et al., 2018), for two samples $\tau, \tau' \sim U([0, 1])$, and policy π , the sampled temporal difference error (TD-Error) at time step t can be computed as:

$$\delta_t^{\tau, \tau'} = r_t + \gamma Z_{\tau'}(x_{t+1}, \pi(x_{t+1})) - Z_{\tau}(x_t, a_t). \quad (4.8)$$

A distributional RL method also comes with loss function. In IQNs, the loss is:

$$L_{dis}(\theta) = \frac{1}{N'} \sum_{i=1}^N \sum_{j=1}^{N'} \rho_{\tau_i}^{\lambda}(\delta_t^{\tau, \tau'}), \quad (4.9)$$

with $\rho_{\tau_i}^{\lambda}$ is the quantile regression term (Dabney et al., 2018), N and N' are the number of samples used to evaluate the TD-error.

4.2.4. Multi-objective robust loss

Figure 4.2 outlines two different reinforcement learning frameworks for learning TSC policies. While distributed RL tends to outperform classical RL in perceptual domains, it's not known how these results might extend to a multi-agent TSC domain.

Early experiments showed important differences between both methods. First, it is found that distributional RL converges faster compared to classical RL in this studied domain. Second, the embeddings learned by two different approaches is compared. In Figure 4.3, t-SNE (Maaten et al., 2008) --- a non-linear dimensionality reduction method --- is used to explore the **deterministic embeddings** ψ and the **distributional embeddings** $\psi \odot \phi$. The same conclusion can also be drawn for each sample in **distributional embeddings**, to visualize the feature, the average across samples is calculated.

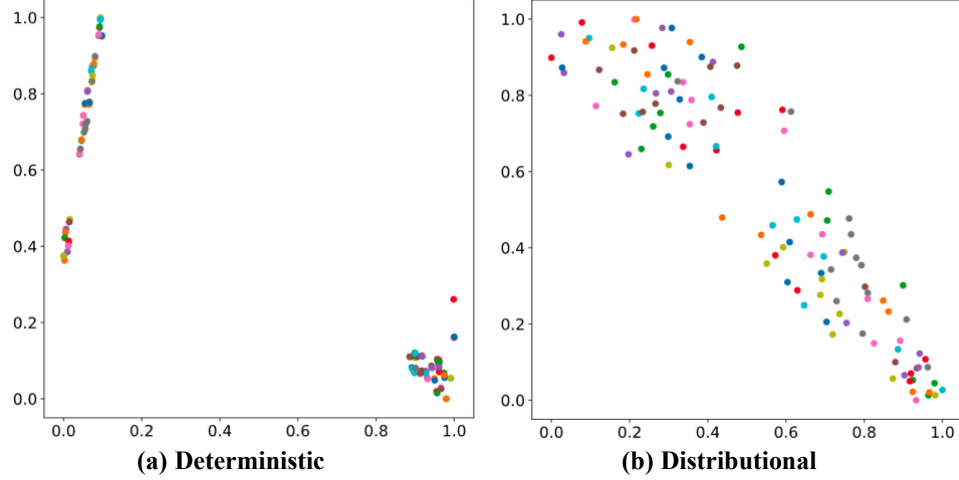


Figure 4.3: Visualization of graph representation feature

The deterministic embeddings form clusters that likely correspond to the two available actions. On the other hand, the distributional embeddings are more evenly distributed, which likely better captures the uncertainty of the actions. The distributional approach uses a robust loss that is less sensitive to outliers and so might discard important information. From the preliminary experiment both methods seem to represent different types of information and so a (convex) combination might yield the best of both worlds.

$$L_{robust} = \kappa L_{MARL} + (1 - \kappa) L_{dis} , (4.10)$$

where $\kappa \in [0,1]$ is the relative importance of the standard RL approach. This new objective can be jointly optimized to improve the robustness of our model. It can be referred to the model that optimizes this loss as distributed graph reinforcement learning (DGRL).

4.3. Experiment analysis

In this section, the effectiveness and interpretability of the proposed distributional RL method for multi-agent TSC is studied. It is aimed to answer the following questions:

- (1) How does the proposed method perform compared with other state-of-the art baselines?
- (2) Is the proposed method robust enough to demand surge and sensor failure problems compared to other baseline methods?
- (3) How to balance the trade-off between representation capacity and the learning stability and improve the overall robustness?

4.3.1. Background and assumptions

The background knowledge and key assumptions for our problem formulation are given as follow:

1) Sensor Failures

In the experiment for TSC, it's assumed that "we can know the lane each vehicle is in". let's imagine that that on each traffic signal controller, there would be a camera/detector that can sense which vehicle has entered into the lane, and it is not likely to fail in reality. As a result, the lane information of each vehicle can be got from the camera.

2) Demand surge

Different traffic demands are based on the arrival rate. The arrival rate is controlled by the option 'period' in Simulation of Urban MObility (SUMO) (Krajzewicz et al., 2002). By default, this generates vehicles with a constant period and arrival rate of $(1/\text{period})$ per second. Note that for different scale of road networks, the same arrival rate will end up with different traffic signal performance. To make a fair comparison, it's considered that the heavy traffic regime as two times the normal traffic regime in simulated data.

In the experiment, normal traffic regime is set as $\text{period}=4$ while heavy traffic regime as $\text{period}=2$.

3) Evaluation metrics

It's considered for several evaluation metrics to compare different methods.

Travel time: The travel time is defined as the time duration between the real departure time and the time the vehicle has arrived. The information is generated for each vehicle as soon as the vehicle arrived at its destination and is removed from the network.

Queue length: The queue length is calculated using the end of the last standing vehicle. This criterion is to measure the congestion.

Delay: The delay is to measure the gap between the vehicle's current speed to the maximum theoretically reachable speed, which is constrained by the type of vehicle and the max allowed speed on its current lane.

$$s_v^* = \min(s_v^*, s_l), \quad (4.11)$$

$$d_t = \sum_{v \in V} (s_v^*, s_v) / s_v^*, \quad (4.12)$$

where V is the total vehicles traveling in the current network, s_v is the maximum speed that the vehicle can reach, s_l is the speed limitation of this road, s_v is the current vehicle speed, finally the delay can be got at time step t .

4.3.2. Experiment setup

The learning setup is shown in Figure 4.4. RL methods (DGRL, IGRL, and GNN-TSC) are trained on synthetic road networks. Then, their performances are tested on either other synthetic networks or perform zero-shot generalization by controlling the TSCs of two real-world networks (a subset of Luxembourg and Manhattan). During training, exploratory behaviors can be encouraged using randomly generated networks. The second advantage of this training scheme is that it doesn't need to re-train the model on the target networks. At test, the effects of missing data and demand surges are studied, which is simulated by using heavier traffic regimes.

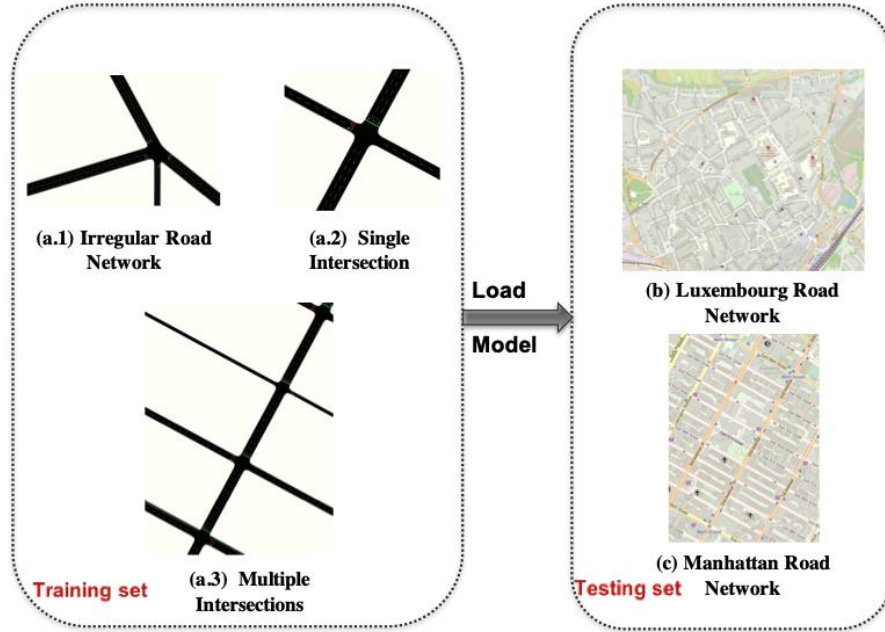


Figure 4.4: Learning scheme for the proposed model

The proposed method is compared with several state-of-the-art methods, including both classical transportation methods and learned ones.

1) Fixed time baseline

This method uses a pre-determined plan for cycle length and phase time, which is widely used when the traffic flow is steady (Koonce et al., 2008).

2) MaxPressure

A popular and strong baseline for network-level traffic signal control method in transportation area. At each time step, it selects the action that maximizes the number of moving vehicles from inbound lanes (Vraiya et al., 2013).

3) Inductive Graph Reinforcement Learning (IGRL)

This recent approach uses graph convolutional networks with a decentralized RL objective. The authors show that their approach can scale and transfer to massive-scale networks. The proposed robust learning framework is based on IGRL. Then, the proposed model is compared

against their best performing model IGRL-V which models vehicles as nodes (Devailly et al., 2020).

4) Graph Neural Networks for TSC (GNN-TSC)

Similar to IGRL, the authors propose a GNN-based RL-trained model. Compared to IGRL (Devailly et al., 2020), the method does not consider individual vehicles as nodes in the graph. Instead, they model information at the lane level. The attention module has also experimented as given in (Wei et al., 2019), but the performance hasn't improved, so it isn't included in the experiment. With that in mind, IGRL-L, a version of IGRL that models lane nodes rather than vehicles as nodes, is used. The authors of (Wei et al., 2019) rely on the CityFlow simulator (Wei et al., 2019); in this thesis, SUMO is used, which makes a direct comparison impossible without a major code rewrite. Independent Reinforcement Learning (IRL)

An independent deep Q-Learning (DQN) agent can be used to model each TSC. DQNs are known to be robust given demand surges and sensor failures (Rodrigues et al., 2019, Zhang et al., 2020). The IRL baseline couples DQNs with recent developments for improved robustness: double Q-Learning (Hasselt et al., 2010), a dueling architecture (Wang et al., 2016), and noisy layers (Fortunato et al., 2017).

4.3.3. Performance comparison

In this section, the performance of the above baselines is compared to the performance of the method proposed in this paper. All experiments are repeated 30 times with different random seeds, and the average results are presented. For every evaluation metric, it has reported the sum of a 1,000 time-step simulation.

1) Comparison under different traffic regime in in Synthetic Networks

Table 4.1 reports the performance of different methods for both normal and heavy traffic regimes in the synthetic network. The demand surge experiment is conducted in a synthetic network because it's hard to control the demand parameter in real networks. In the experiment,

train and test networks are disjoint. The same road network is conducted (not seen in the training set) in the test for all methods with 30 random seeds for trips. The distributional RL approach (DGRL) outperforms others in both regimes across the three metrics. DGRL shines in heavy regime showing that it is more robust to demand surges.

It can be seen that Fixed time does not perform as well as MaxPressure in normal traffic regimes but better than MaxPressure in heavy traffic regimes. This suggests that MaxPressure is likely to end up with locally optimal solutions. In terms of travel time, DGRL is the same as IGRL in the normal regime. In a given situation, the average travel time of DGRL is sometimes longer than IGRL's, but the DGRL's trip distribution is in a more equitable fashion with less variability for the same trip. In a heavy traffic regime, it can be seen that DGRL outperforms IGRL by a large margin.

Table 4.1: Comparison under different traffic regime

Methods	Normal regime			Heavy regime		
	Delay	Queue length	Travel time	Delay	Queue length	Travel time
Fixed time	789.26 (± 36.36)	588.88 (± 35.39)	1182.26 (± 125.57)	4059.19 (± 108.54)	4553.34 (± 112.34)	13901.72 (± 922.15)
Max pressure	379.91 (± 12.22)	191.91 (± 10.41)	670.28 (± 264.48)	6201.11 (± 183.23)	6865.94 (± 190.42)	15150.86 (± 734.36)

IQL	1257.58 (± 31.84)	1013.89 (± 29.40)	1242.38 (± 46.78)	5257.58 (± 152.62)	6670.75 (± 160.25)	14112.98 (± 498.12)
GNN- TSC	311.85 (± 4.32)	210.43 (± 10.53)	517.15 (± 34.32)	2998.63 (± 61.47)	3645.75 (± 92.68)	6092.63 (± 428.75)
IGRL	288.16 (± 8.66)	125.89 (± 7.72)	501.36 (± 22.22)	2962.92 (± 81.81)	3515.23 (± 86.00)	6051.32 (± 355.51)
DGRL	244.15 (± 4.25)	80.11 (± 2.74)	501.95 (± 20.77)	2503.96 (± 71.91)	3029.45 (± 76.57)	5030.31 (± 313.82)

2) Comparison under sensor failures in different real-world road networks

In this experiment, the proposed model's performance is tested with two real-world road networks using real traffic demand. The IQL method does not scale to such large networks (the parameters increase linearly with the number of TSCs), and so it cannot be reported for its performance. Transportation baselines do not consider speed nor vehicle position, and so their performance is robust to noisy sensors.

The performance in Manhattan road network is reported in Table 4.2. Missing probabilities 20%, 40%, 60% are evaluated. Because the fixed time and MaxPressure baselines don't use the vehicle's speed and position information, so they won't be affected by missing values in our experiment.

Interestingly, it can be seen that if considering a small missing probability, i.e., 20% into the training set, the performance increases. This finding is in accordance with the inductive ability of GNN to infer for un-sampled speed and position information (Wu et al., 2020). Furthermore, with the consideration of modeling distribution, the training will be more stable given a higher missing probability compared to other RL baselines. As a result, the DGRL model's performance will decrease less than the IGRL model and GAT-TSC model when increasing the missing probabilities.

Another road network with more irregular roads is selected to evaluate the performance of our model. Overall, DGRL outperforms other methods. In Table 4.2, it can be seen that MaxPressure performs worse than the Fixed time method, which demonstrates that when the traffic conditions become more realistic, MaxPressure tends to fail. Furthermore, given higher missing probabilities, i.e., 60%, both IGRL, and GAT-TSC will perform worse than Fixed time method, which suggests that these methods are not robust under higher missing probabilities. Note that the IQL cannot be generalized into large-scale network, so the results are not reported.

Table 4.2: Comparison under different missing values in Manhattan network

Methods	Missing probability (20/40/60%)		
	Delay	Queue length	Travel time
Fixed time	1356.45(± 41.29)	937.47 (± 40.48)	1871.86 (± 238.99)
Max pressure	1144.30 (± 34.32)	907.24 (± 44.43)	1630.67 (± 264.48)
IQL	-	-	-
GNN-TSC	484.49 (± 4.84)/	469.75 (± 7.84)/	973.46 (± 27.23)/
	497.18 (± 9.61)/	578.98 (± 9.61)/	1273.31 (± 12.67)/
	696.15 (± 9.82)	696.15 (± 9.82)	1346.75 (± 41.45)

IGRL	413.94 (± 9.94)/	314.74 (± 3.96)/	966.65 (± 25.47)/
	518.41 (± 11.87)/	417.93 (± 3.36)/	1163.89 (± 10.32)/
	653.22 (± 13.76)	499.89 (± 3.55)	1260.46 (± 18.27)
DGRL	364.23 (± 3.95)/	311.99 (± 3.01)/	954.28 (± 15.66)/
	397.91 (± 4.05)/	363.60 (± 3.17)/	1032.58 (± 13.63)/
	492.89 (± 9.12)	403.11 (± 3.22)	1088.67 (± 17.3)

Table 4.3: Comparison under different missing values in Luxembourg network

Methods	Missing probability (20/40/60 %)		
	Delay	Queue length	Travel time
Fixed time	594.22 (± 16.24)	509.79 (± 14.33)	620.98 (± 68.54)
Max pressure	754.27 (± 22.16)	661.03 (± 19.97)	781.38 (± 131.84)
IQL	-	-	-
GNN-TSC	489.50 (± 6.38)/	385.65 (± 5.06)/	534.16 (± 29.69)/
	595.84 (± 8.82)/	511.68 (± 8.71)/	651.36 (± 49.48)/
	723.65 (± 10.79)	627.66 (± 10.59)	721.98 (± 58.02)
IGRL	438.26 (± 8.31)/	373.33 (± 4.89)/	527.38 (± 31.20)/
	531.25 (± 9.30)/	460.07 (± 6.23)/	591.92 (± 32.71)/
	678.75 (± 14.37)	589.61 (± 7.35)	683.25 (± 40.51)
DGRL	419.43 (± 6.23)	356.28 (± 3.27)	467.94 (± 16.35)
	501.86 (± 7.12)/	421.85 (± 5.71)/	535.66 (± 23.98)/
	545.68 (± 8.56)	469.28 (± 7.91)	572.67 (± 28.01)

As in Table 4.2 and Table 4.3, DGRL and IGRL have similar performance in normal traffic regime with 20% missing probability, respectively. Then, the same trips' travel time is compared between DGRL and other representative baselines.

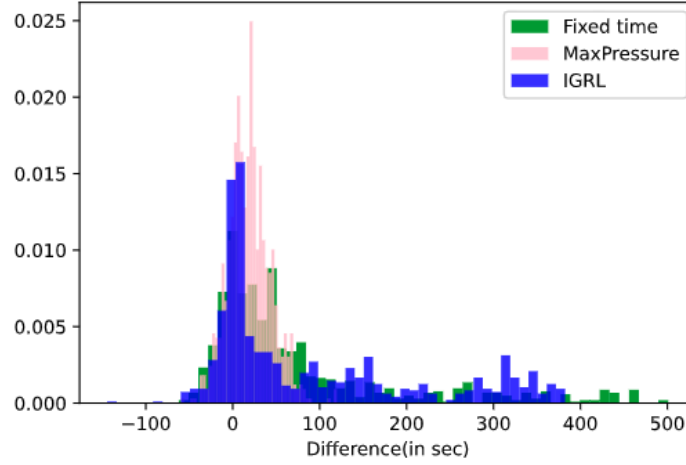


Figure 4.5: Trip duration comparison

In Figure 4.5, differences of paired trips travel time compared to DGRL. The difference between DGRL and the method is reported (i.e., $DGRL - \text{method}$). The numbers higher than 0 indicate the method being outperformed by DGRL. The y-axis is normalized.

As shown in Figure 4.5, although IGRL and DGRL have similar average trips, lots of trip are delayed by IGRL, e.g., 100--200 and 300--400, so we conclude that DGRL distributes these trip delays much more smoothly.

To visualize the control performance, the average delays per time step is collected from two road networks. The best RL baseline is selected and two transportation baselines to make a clear comparison.

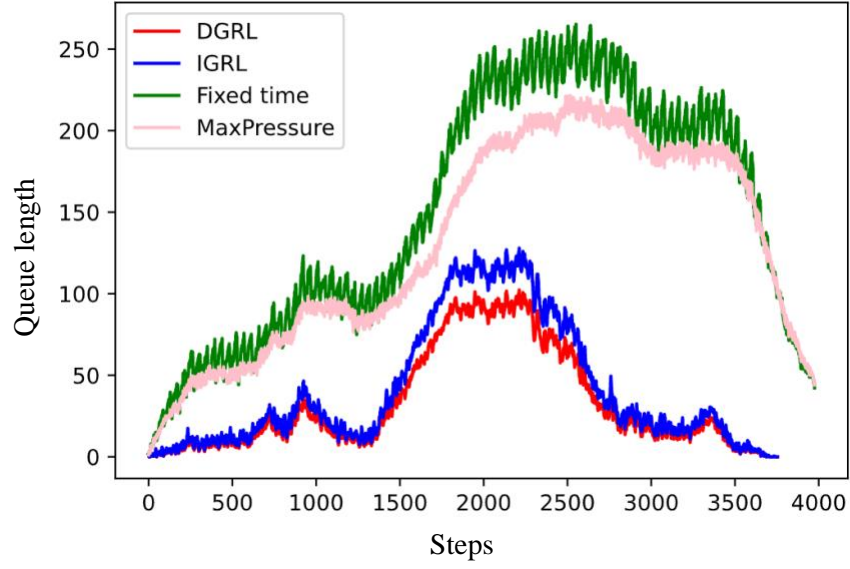


Figure 4.6: Average delays evolution in Manhattan road network

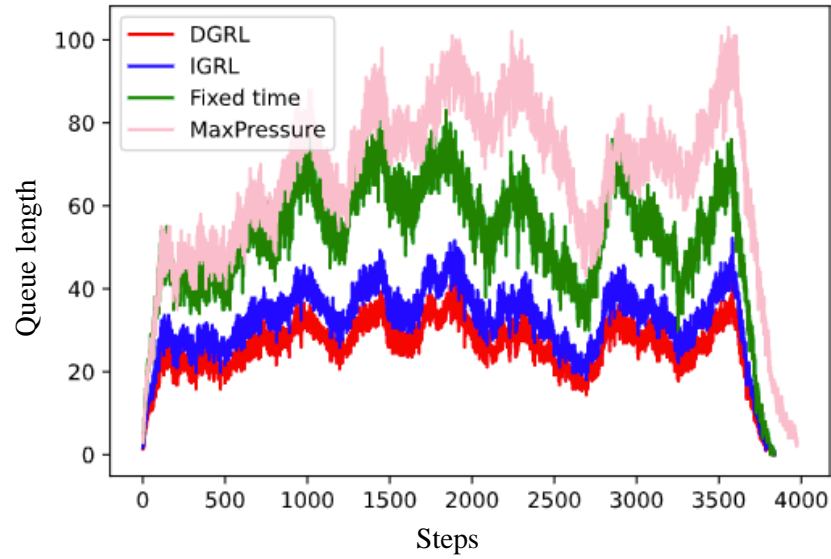


Figure 4.7: Average delays evolution in Luxembourg road network

From Figure 4.6, it can be seen that DGRL better mitigates the effect of demand surge compared to other baselines. Moreover, from Figure 4.7, with more challenging demand evolution in Luxembourg road network, DGRL also demonstrates the overall best robustness.

3) Comparison with Different Weight Parameter in Loss Function

Furthermore, the weight parameter κ in the loss function is evaluated; it is an important parameter to balance between the representation capacity and the learning stability. In Figure 4.8, it can be seen that either model on their own ($\kappa = 0$ or $\kappa = 1$) never perform as well as their combination. When κ is close to 1 (MARL) or 0 (Dis), which suggests that the model cannot perform well with only distributional RL and deep graph RL.

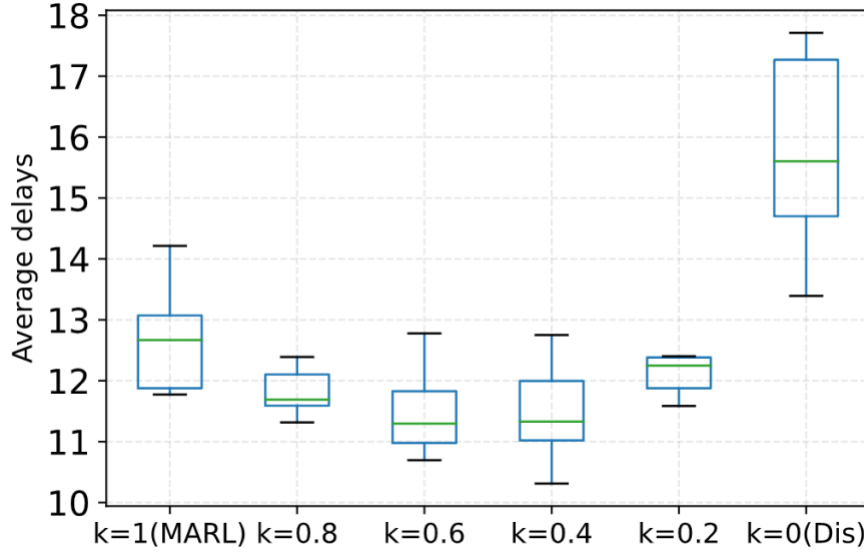


Figure 4.8: Different κ combination

Also, it can be found that $\kappa=0$ is worse than 1, it validates that modeling the distribution may end up losing some important features. On the other hand, with the combination of these two objectives, the model can be quite robust by striking the trade-off between the representation capacity and the learning stability.

4) Model architecture analysis

Number of samples is an important hyper-parameter for model's performance (Dabney et al., 2018). Different number of samples have been tested, it's found that although with larger samples, the performance in previous few episodes would be better, however, just a minimal impact on

long performance. Furthermore, more samples will make the training become harder, harming the overall performance, as a result, the $N=8$ is used in our experiment.

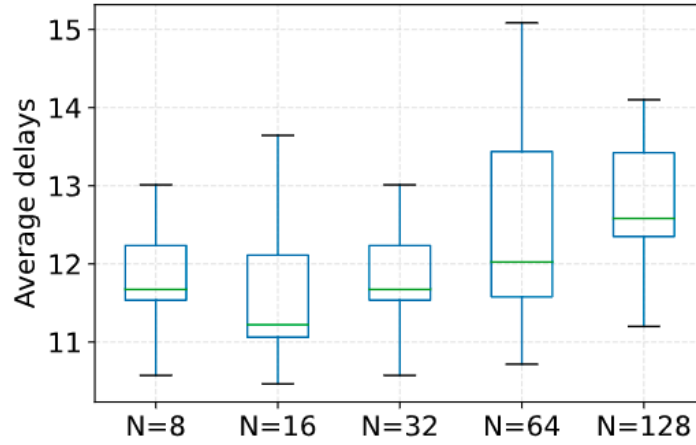


Figure 4.9: Different number of samples

To evaluate the learning stability of learning distribution versus learning deterministic values, the value estimation is recorded from the value function. For the distributional value function, the average value is calculated over the number of samples to compare with the deterministic value. From Figure 4.10, it can be found that learning distribution can converge faster, which demonstrates that estimate the distribution will present a more stable learning effect. Then, the same conclusion can also be drawn from both scenario with and without exogenous uncertainty.

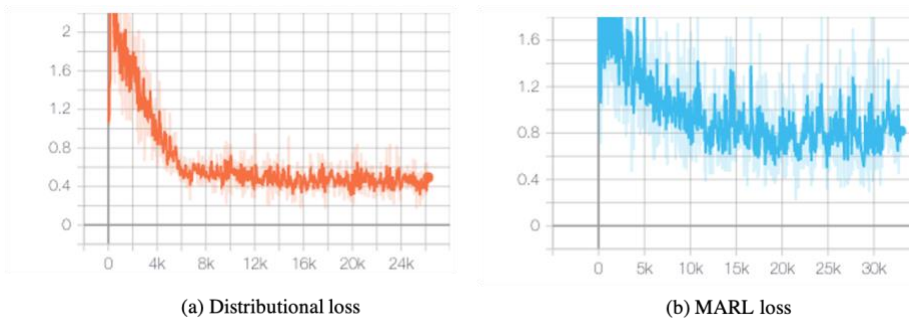


Figure 4.10: Comparison of convergence

4.4. Main findings

In this chapter, it is demonstrated that with the help of distributional reinforcement learning, the proposed method is particularly robust to exogenous uncertainty. This is done by comparing traditional methods, fixed time and MaxPressure methods and using the networks of Luxembourg and Manhattan road networks. However, the naive model fails when considering demand surge or sensor failure problems. The proposed method also can enable a flexible trade-off to improve the overall decision performance and system robustness, achieving the best performance in all the designed scenarios. Furthermore, policies learned with DGRL can also enable both transfer and scaling ability to large-scale networks.

Chapter 5. Conclusion and Future Work

5.1. Conclusions

In the first part of the CAV research, this thesis focus on encouraging cooperation in such mixed autonomy scenario. The proposed graph convolutional reinforcement learning approach is efficient for CAV control by encouraging cooperation through information aggregation. The main contribution is that graph attention network is first use in the mixed autonomy setting to capture mutual interplay in order to encourage agent cooperation.

Extensive experiments are conducted 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, 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 thesis, can significantly enhance the cooperation among agents, which improves both efficiency and safety of the system. Overall, the proposed method can achieve the overall best performance under different road networks, target speeds, penetration rates. These findings provide valuable insights into the design of the connected and automated driving system.

In the second part of TSC control research, this thesis focus on further improving the robustness to exogenous uncertainty based on the graph reinforcement learning in the first part. The main contribution is the proposed method can achieve a flexible trade-off to improve overall decision performance and robustness to exogeneous uncertainty.

An RL approach is proposed based on Distributional Graph Reinforcement Learning (DGRL) for large-scale traffic signal control. DGRL is particularly robust to exogenous uncertainty. This is the first study on how to consider robustness in large-scale TSC as well as integrate graph neural networks with distributional reinforcement learning in multi-agent settings. Furthermore, policies learned with DGRL can enable both transfer and scaling ability to large-scale networks. A series of experiments are conducted on two different real-world networks with

real traffic demands and show that the proposed method outperforms several state-of-the-art baselines.

5.2. Future work

In the first part of the CAV research, 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, it is also worthy of trying to develop sim-to-real transfer learning (Jang et al., 2019) for mixed-autonomy control and implement our approach in real mobile robot vehicles. Fairness is also important for modern society, which can contribute to the stability and productivity of the multi-agent system. As in the mixed autonomy system, human-driving vehicles and automated vehicles should maintain fairness and envy-freeness reward. To tackle this challenge, the hierarchical reinforcement learning model can be investigated in the future (Jiang. et al., 2019).

In the second TSC control research, it is interested in studying the empirical and theoretical properties of DGRL to robustly model other multi-agent systems with exogenous sources of uncertainty. It is valuable to evaluate the effects of various sampling distribution in Implicit Quantile Networks (Dabney et al., 2018). Furthermore, as shown in Figure 4.3, different distribution is corresponding to different action selection. In the future, it's also helpful to investigate which kind of distributions are corresponding to which kind of actions.

It is promising to study intelligent transportation with both intelligent traffic signal control and connected vehicles together. In such a system, the traffic signal control system guide the movements of both human-driving vehicle and connected vehicle while the connected vehicle will also help regulate traffic flow. It is interesting to develop a unified control framework for both traffic signals and connected vehicles.

References

- M. Treiber and A. Kesting, “Traffic flow dynamics,” Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg, 2013.
- 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.
- 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.
- C. Wu, A. Kreidieh, K. Parvate, E. Vinitzky, and A. M. Bayen, “Flow: Architecture and benchmarking for reinforcement learning in traffic control,” arXiv preprint arXiv:1710.05465, 2017.
- 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.
- François-Xavier Devailly, Denis Larocque, and Laurent Charlin. 2020. “IG-RL: Inductive Graph Reinforcement Learning for Massive-Scale Traffic Signal Control,” arXiv preprint arXiv:2003.05738(2020).
- Tianyu Shi, Pin Wang, Xuxin Cheng, Ching-Yao Chan, and Ding Huang. 2019. “Driving decision and control for automated lane change behavior based on deep reinforcement learning,” In 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2895–2900.
- Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. “Intellilight: A reinforcement learning approach for intelligent traffic light control,” In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2496–2505.
- Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. “Colight: Learning network-level cooperation for traffic signal control,” In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 1913–1922.

Qureshi, K. N., & Abdullah, A. H. 2013. "A survey on intelligent transportation systems," Middle-East Journal of Scientific Research, 15(5), 629-642.

Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. 2020. "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," IEEE Transactions on Vehicular Technology, 69(12), 14413-14423.

Yang, Y., & Wang, J. (2020). "An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective," arXiv preprint arXiv:2011.00583.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. 2017. "Multi-agent actor-critic for mixed cooperative-competitive environments," arXiv preprint arXiv:1706.0227.

S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," arXiv preprint arXiv:1610.03295, 2016.

P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," arXiv preprint arXiv:1911.04175, 2019.

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

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.

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.

Park, J., Min, K., & Huh, K. (2019, December). "Multi-Agent Deep Reinforcement Learning for Cooperative Driving in Crowded Traffic Scenarios," In 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) (pp. 1-2). IEEE.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., & Wang, J. (2018, July). "Mean field multi-agent reinforcement learning," In International Conference on Machine Learning (pp. 5571-5580). PMLR.

- Bacchiani, G., Molinari, D., & Patander, M. 2019. “Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning,” arXiv preprint arXiv:1903.01365.
- Jiang, J., & Lu, Z. 2019. “Learning fairness in multi-agent systems,” arXiv preprint arXiv:1910.14472.
- 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.
- 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.
- S. Iqbal and F. Sha, “Actor-attention-critic for multi-agent reinforcement learning,” in International Conference on Machine Learning, 2019, pp.2961–2970.
- A. Agarwal, S. Kumar, and K. Sycara, “Learning transferable cooperative behavior in multi-agent teams,” arXiv preprint arXiv:1906.01202, 2019.
- 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.
- 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. 1475–148.
- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, 2017.
- E. Vinitzky, 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.
- 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.
- 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.

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.

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.

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.

K. Jang, E. Vinitzky, B. Chalaki, B. Remer, L. Beaver, A. A. Malikopoulou, 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.

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. 2020. “An optimistic perspective on offline reinforcement learning,” In International Conference on Machine Learning. PMLR, 104–114.

Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. “A distributional perspective on reinforcement learning,” arXiv preprint arXiv:1707.06887(2017).

Chen Cai, Chi Kwong Wong, and Benjamin G Heydecker. 2009. “Adaptive traffic signal control using approximate dynamic programming,” Transportation Research Part C: Emerging Technologies 17, 5 (2009), 456–474.

Xinyu Chen, Zhaocheng He, Yixian Chen, Yuhuan Lu, and Jiawei Wang. 2019. “Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model,” Transportation Research Part C: Emerging Technologies 104 (2019), 66–77.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. “Multi-agent deep reinforcement learning for large-scale traffic signal control,” IEEE Transactions on Intelligent Transportation Systems 21, 3 (2019), 1086–1095.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. “Implicit quantile networks for distributional reinforcement learning,” arXiv preprint arXiv:1806.06923(2018).

Mohamed Essa and Tarek Sayed. 2020. “Self-learning adaptive traffic signal control for real-time safety optimization,” *Accident Analysis & Prevention* 146(2020), 105713.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. 2017. “Noisy networks for exploration,” *arXiv preprint arXiv:1706.10295*(2017).

Hado V Hasselt. 2010. “Double Q-learning,” In *Advances in neural information processing systems*. 2613–2621.

Peter Koonce and Lee Rodegerdts. 2008. “Traffic signal timing manual,” Technical Report. United States. Federal Highway Administration.

Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. 2002. “SUMO (Simulation of Urban MObility)-an open-source traffic simulation,” In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*. 183–187.

Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. 2020. “Multi-agent game abstraction via graph attention neural network,” In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7211–7218.

Daniel J Lizotte, Lacey Gunter, Eric Laber, and Susan A Murphy. 2008. “Missing data and uncertainty in batch reinforcement learning,” In *Neural Information Processing Systems (NIPS)*

Clare Lyle, Marc G Bellemare, and Pablo Samuel Castro. 2019. “A comparative analysis of expected and distributional reinforcement learning,” In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4504–4511.

Laurens van der Maaten and Geoffrey Hinton. 2008. “Visualizing data using t-SNE,” *Journal of machine learning research*, Nov (2008), 2579–2605.

Tien Mai, Quoc Phong Nguyen, Kian Hsiang Low, and Patrick Jaillet. 2019. “Inverse Reinforcement Learning with Missing Data,” *arXiv preprint arXiv:1911.06930*(2019).

Filipe Rodrigues and Carlos Lima Azevedo. 2019. “Towards Robust Deep Reinforcement Learning for Traffic Signal Control: Demand Surges, Incidents and Sensor Failures,” In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 3559–3566.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. “Mastering the game of go without human knowledge,” *nature* 550,7676 (2017), 354–359.

Pravin Varaiya. 2013. “The max-pressure controller for arbitrary networks of signalized intersections,” In *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, 27–66.

Jiawei Wang, Tianyu Shi, Yuankai Wu, Luis Miranda-Moreno, and Lijun Sun. 2020. “Multi-agent Graph Reinforcement Learning for Connected Automated Driving,” In *Proceedings of International Conference on Machine Learning (ICML) Workshop on AI for Autonomous Driving*.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. “Dueling network architectures for deep reinforcement learning,” In *International conference on machine learning*. PMLR, 1995–2003.

MA Wiering, J van Veenen, Jilles Vreeken, and Arne Koopman. 2004. “Intelligent traffic light control”.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. “Graph neural networks: A review of methods and applications,” *arXiv preprint arXiv:1812.08434*(2018).

Appendix

For the Luxembourg road network in Figure A.1, it has two peak hours, while for the Manhattan road network, as given in Figure A.2, it only has one peak hour. The configuration of different road networks is shown in Table A.1.

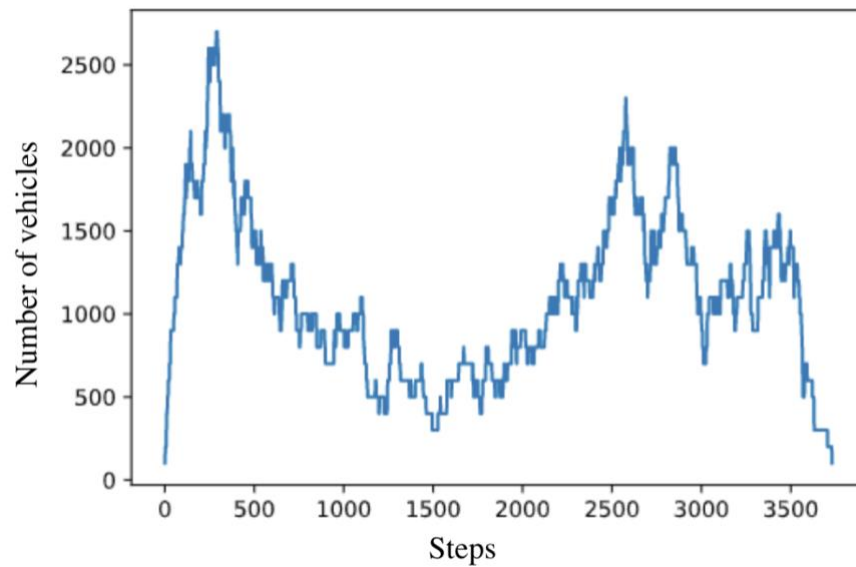


Figure A.1: Traffic demand evolution in Luxembourg road network

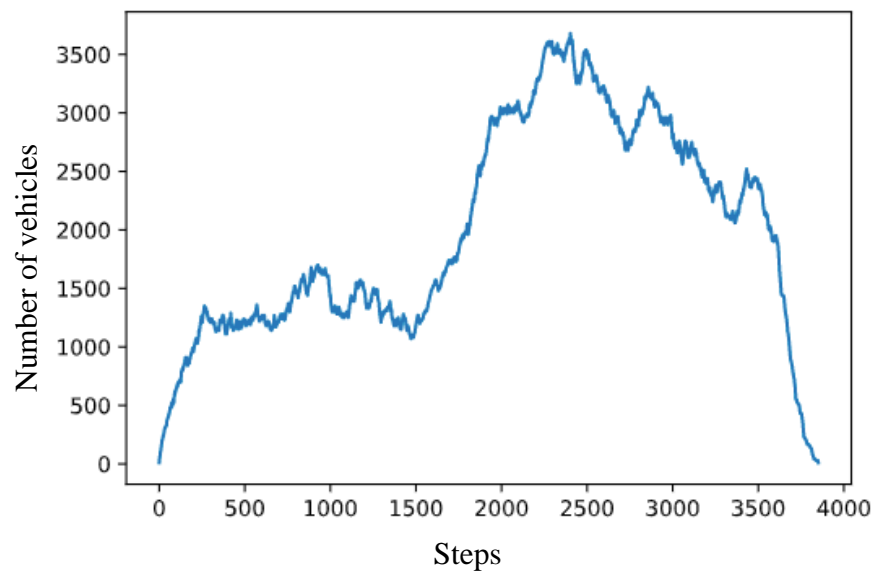


Figure A.2: Traffic demand evolution in Manhattan road network

Table A.1: Configuration of different road networks

Road network	Traffic light	Number of intersections
Luxembourg	75	550
Manhattan	22	482