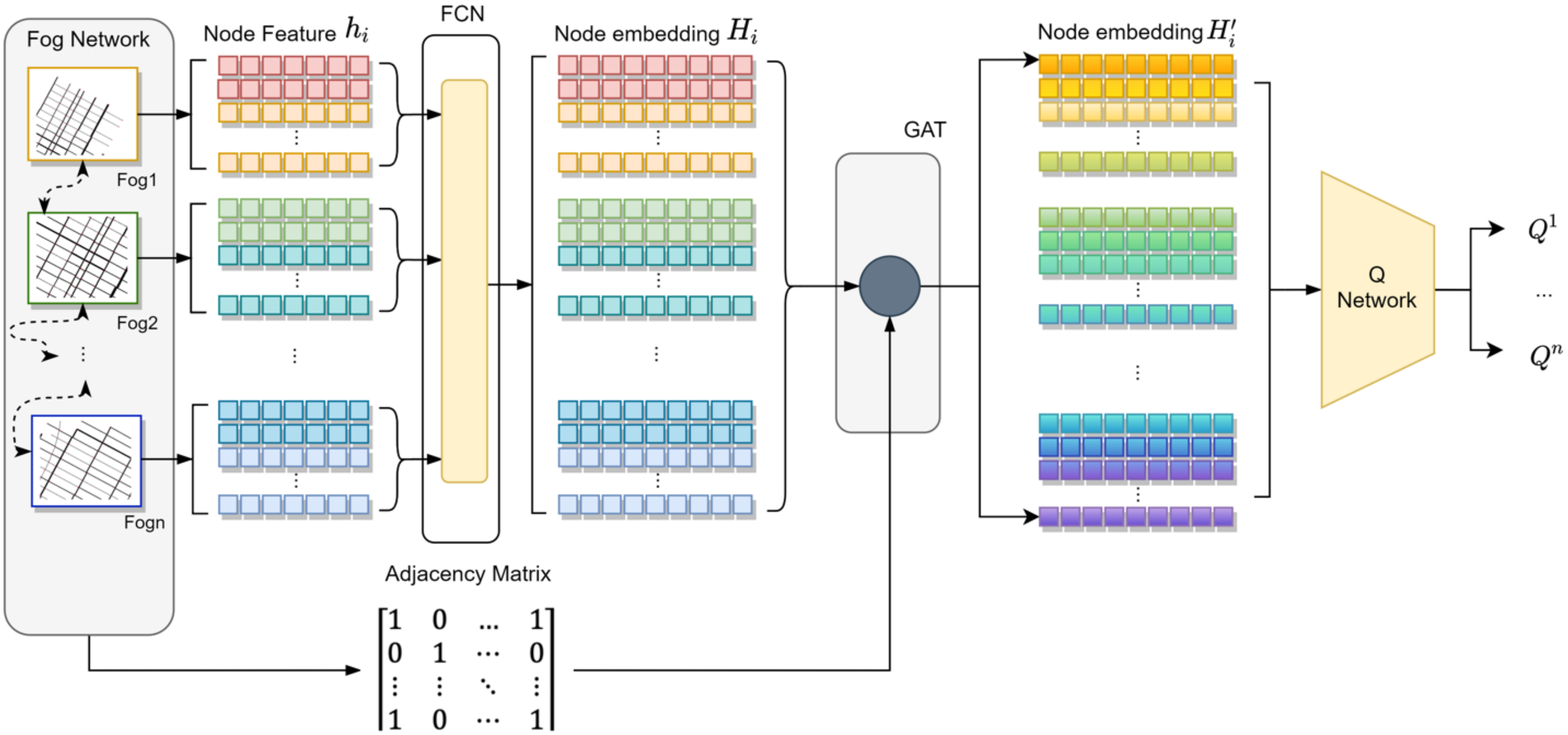
CENTER FOR CONNECTED AND
AUTOMATED TRANSPORTATION

Final Report #55
June 2022



Large network multi-level control for CAV and Smart Infrastructure: AI-based Fog-Cloud collaboration

Runjia Du
Paul (Young Joun) Ha
Jiqian Dong
Sikai Chen
Samuel Labi

PURDUE
UNIVERSITY

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Report No. 55
Project Start Date: January 2021
Project End Date: December 2021

June 2022

Large network multi-level control for CAV and Smart Infrastructure: AI-based Fog-Cloud collaboration

Runjia Du
Graduate Researcher

Paul Young Joun Ha
Graduate Researcher

Jiqian Dong
Graduate Researcher

Sikai Chen
Visiting Asst. Professor

Samuel Labi
Professor

Purdue University

ACKNOWLEDGEMENTS AND DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. 69A3551747105 of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested APA Format Citation:

Du, R., Ha, P.Y.J., Dong, J., Chen, S., Labi, S. (2022). Large network multi-level control for CAV and Smart Infrastructure: AI-based Fog-Cloud collaboration, CCAT Report #55, The Center for Connected and Automated Transportation, Purdue University, West Lafayette, IN.

Contacts

For more information:

Samuel Labi, Ph.D.
550 Stadium Mall Drive
HAMP G167B
Phone: (765) 494-5926
Email: labi@purdue.edu

CCAT
University of Michigan Transportation
Research Institute
2901 Baxter Road
Ann Arbor, MI 48152

uumtri-ccat@umich.edu
(734) 763-2498
www.ccat.umtri.umich.edu

Technical Report Documentation Page

1. Report No. 55	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Large network multi-level control for CAV and Smart Infrastructure: AI-based Fog-Cloud collaboration		**5. Report Date** June 2022 **6. Performing Organization Code** N/A
7. Author(s) Runjia Du, Paul Y.J. Ha, Jiqian Dong, Sikai Chen, Samuel Labi		**8. Performing Organization Report No.** N/A
9. Performing Organization Name and Address Center for Connected and Automated Transportation Purdue University, 550 Stadium Mall Drive, W. Lafayette, IN 47907; and Univ. of Michigan Ann Arbor, 2901 Baxter Rd, Ann Arbor, MI 48109		**10. Work Unit No.** **11. Contract or Grant No.** Contract No. 69A3551747105
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE, Washington, DC 20590		**13. Type of Report and Period Covered:** Final rep., Jan 2021-Dec 2021 **14. Sponsoring Agency Code:** OST-R

15. Supplementary Notes

Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's (OST-R) University Transportation Centers (UTC) program.

16. Abstract

The first part of this study addresses the use of fog-cloud architecture for a deep reinforcement learning-based control framework and presents a case study involving urban traffic dynamic rerouting. Past work has shown that dynamic rerouting can mitigate traffic congestion and can be facilitated using emerging technologies such as Deep Reinforcement Learning (DRL) and fog-computing. However, two unaddressed challenges include the immense size of the action space associated with urban road networks, and the impairment of learning efficiency engendered by the large size of THE network information. Therefore, this project proposes a two-stage model that combines GAQ (Graph Attention Network – Deep Q Learning) and EBkSP (Entropy Based k Shortest Path) overlying a fog-cloud information architecture, for higher learning efficiency by shrinking action space and selecting relatively important information to reroute vehicles in a dynamic urban environment. First, the GAQ analyzes the traffic conditions and EBkSP assigns a route to each vehicle based on two criteria. Using a case study, the proposed model is tested and the results demonstrate the efficacy of the model for rerouting vehicles in a dynamic manner. The second part of the study uses fog-cloud based multiagent reinforcement learning scalable for controlling a specific class urban transport systems – traffic signal systems. Optimizing traffic signal control (TSC) at intersections continues to pose a challenging problem, particularly for large-scale traffic networks. While it is feasible to optimize the operations of individual TSC systems or a small number of such systems, it is computationally difficult to scale these solution approaches to large networks partly due to the curse of dimensionality that is encountered as the number of intersections increases. Fortunately, recent studies have recognized the potential of machine learning tools address this problem. However, facilitating such intelligent solution approaches may require unduly large investments in infrastructure such as roadside units (RSUs) and drones in order to ensure thorough connectivity across all intersections in large networks, an investment that may be financially burdensome to road agencies. As such, this study builds on recent work to present a scalable TSC model that may reduce the number of required enabling infrastructure in this problem context. This study uses graph attention networks (GATs) to serve as the neural network for deep reinforcement learning, which aids in maintaining the graph topology of the traffic network while disregarding any irrelevant or unnecessary information. A case study is carried out to demonstrate the effectiveness of the proposed model, and the results show much promise. The overall research outcome suggests that by decomposing large networks using fog-nodes, the proposed fog-based graphic RL (FG-RL) model can be easily applied to scale into larger traffic networks.

17. Key Words Autonomous vehicles, Infrastructure, Multiagent reinforcement learning, Traffic signal control, Graph neural networks	**18. Distribution Statement** No restrictions.

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 56	**22. Price**

TABLE OF CONTENTS

LIST OF TABLES……………………………………………………………… 4

LIST OF FIGURES…………………………………………………………….. 6

LIST OF ACRONYMS………………………………………………………… 7

PART 1

CHAPTER 1 INTRODUCTION…………………………………………………….. 8
1.1. Background……………………………………………………………………... 8
1.2. Congestion-mitigation palliatives………………………………………………. 8
1.3. Congestion reduction using vehicle rerouting ………………………………. 9
1.4. Overview and organization of the project………………………………………. 9

CHAPTER 2 UNDERLYING CONCEPTS OF THE PROPOSED FRAMEWORK 10
2.1. Prospective role of fog-cloud collaboration in vehicle rerouting…………………… 10
2.2. Attention mechanism for deep reinforcement learning …………………………… 11
2.3. Discussion……………………………………………………………………… 12

CHAPTER 3 PROBLEM SETTINGS………………………………………………… 13
3.1. DRL stage……………………………………………………………………… 14
3.2. Route-assignment stage………………………………………………………… 15

CHAPTER 4 METHODOLOGY……………………………………………………… 15
4.1. DRL model architecture………………………………………………………… 15
4.2. Routes assigned model architecture……………………………………………….. 18

CHAPTER 5 EXPERIMENT SETTINGS…………………………………………….. 19
5.1. Simulator parameters…………………………………………………………… 19
5.2. Baseline models………………………………………………………………… 21

CHAPTER 6 RESULTS……………………………………………………………… 21
6.1. Training stage…………………………………………………………………... 21
6.2. Testing stage…………………………………………………………………… 24

CHAPTER 7 CONCLUSIONS………………………………………………………… 26

PART II

CHAPTER 8: INTRODUCTION ……………………………………………………… X

CHAPTER 9. BACKGROUND FOR THE METHODOLOGY …………………….. X

9.1 Reinforcement Learning…………………………………………………………… X
9.2 Graph Neural Networks………………………………………………………… X

CHAPTER 10. METHODOLOGY ………………..…………………………………….. X
10.1 DRL Model Architecture ……………………………………………………... X

CHAPTER 11. CASE STUDY ………………………………………………………… X
11.1 Network Descriptions ……………………………………………………….. X
11.2 MDP Settings ……………………………………………………………………... X
11.3 Preliminary Results …………………………………………………………….. X

CHAPTER 12 CONCLUSIONS ………………..…………………………………….. X

CHAPTER 13 STUDY OUTCOMES AND OUTPUTS ………………………………. X
12.1 Outputs ……………………………………………………………………………..X
12.3 Outcomes ………………………………………………………………………….. X
12.4 List of Impacts …………………………………………………………………….. X

REFERENCES ……………………………………………………………………………X

APPENDIX ……………………………………………………………………………..X

LIST OF TABLES

Table 1 Training performance comparison (episode reward) with different parameters 29

Table 2 Training performance comparison (episode reward) for different model 29

LIST OF FIGURES

Figure 1 Fog-cloud collaboration.. 14

Figure 2 Fog-node graph structure.. 15

Figure 3 Dynamic vehicle rerouting framework architecture 16

Figure 4 Illustration of BVs (white) and RVs (green) on the road network 16

Figure 5 Logical flow of the route assignment stage 19

Figure 6 DRL model architecture 20

Figure 7 Network used in experiments 26

Figure 8 Training performance of different priority standards 28

Figure 9 Reward comparison (proposed vs. balanced).. 29

Figure 10 Testing performance of rule-based and RL-based models under different scenarios .. 31

Figure 11: GAT model architecture ………………………………………………………….X

Figure 12: Small TSC Network ……………………………………………………………….X

Figure 13: Comparison of training results using 2 fog nodes vs fully observable system ………X

Figure 14: Average intersection delay ………………………………………………………...X

LIST OF ACRONYMS

BV – Background Vehicles, that is, vehicles that are not rerouted but are incorporated in the framework to add randomness and dynamics in the network

DGN – Graph Convolutional Reinforcement Learning

DRL – deep reinforcement learning

EBkSP – Entropy Balanced k Shortest Path algorithm

FG-RL model – Fog-based Graphic RL model

GAT – Graph Attention Network

GCQ – Graphic Convolutional Q Learning

GCN – Graph Convolutional Neural Network

GNN – Graph Neural Network

GPU – Graphics Processing Unit

HDV – Human-Driven Vehicles

ITS – Intelligent Transportation Systems

MARL – Multiagent Reinforcement Learning

MDP – Markov Decision Process

OSM – Open Street Map

RV – Rerouting Vehicles

TMC – Transportation Management Center

TSC – Traffic Signal Control

V2V – Communication between vehicles and other vehicles

VMS – Variable Message Signs

Vram – Video Random Access Memory

PART I

Using Fog-Cloud Architecture for a Deep Reinforcement Learning-based Control Framework and Case Study involving Urban Traffic Dynamic Rerouting

CHAPTER 1 INTRODUCTION

1.1. Background

In the current era of rapid urbanization and motorization, traffic congestion continues to impair urban travel experience and quality of life. According to the United Nations, 54 percent of the world's population resides in urban areas, and this percentage will rise to 66 percent by 2050 (United Nations 2014, 2015). Both developing and developed countries are experiencing growth in vehicle ownership or use, with 1.4 billion vehicles globally in 2021. In the United States, for example, the total number of registered vehicles increased from 250 million to 273 million in the 8-year period between 2010 and 2018 (U.S. Department of Transportation, 2020). There is a myriad of demand-side and supply-side palliatives for congestion reduction, as we discuss in the next subsection of this report. However, some of these that initially offered much promise, including rideshare and delivery services, rather exacerbated the congestion problem than cure it, because they (paradoxically) led to travel demand increase (INRIX, 2020).

1.2. Congestion-mitigation palliatives

Generally, urban traffic congestion occurs when traffic flow exceeds road capacity and is generally addressed through any (or a combination) of three categories of mitigation palliatives: (a) adding more capacity, (b) promoting travel and land-use patterns reduce or flatten demand, (c) using the existing capacity more efficiently (U.S Department of transportation, 2012). The first category of palliatives includes capacity expansion through new corridors or additional travel lanes. However, it is well known that this may not always yield the intended benefits due to induced demand (Karimi et al., 2021). With regard to the second category, classic initiatives include staggered work hours, work-from-home, carpooling, congestion pricing, and other demand reduction strategies. Of these, congestion pricing (CP) seems to have the greatest potential to reduce congestion. CP harnesses the power of the market to reduce congestion. However, CP may lead to inequitable outcomes (Eliasson et al., 2006). With regard to the third category, there has been much promise of using advanced technologies and real-time information to mitigate congestion. With the development of the emerging technologies including connectivity and automation, Intelligent Transportation Systems (ITS) and real-time control through Transportation Management Center (TMC) are becoming increasingly feasible. Connectivity, in particular, plays an important role in the process of acquiring and using real-time traffic information to enhance travel efficiency. Through connectivity, vehicles are able to communicate with other vehicles (V2V) and infrastructures (V2I) (Nguyen et al., 2020, 2021). Further, mobile technologies, such as smartphones and Bluetooth, which are convenient and affordable, can be used to provide V2V connectivity services. In addition, V2V connectivity is not susceptible to occlusion or inclement weather, and thus offers high accuracy of information with fewer limitations. Recognizing the merits of connectivity in terms of reliability and affordability, vehicle manufacturing and technology companies seek ways to install connectivity-enabling devices on human-driven vehicles (HDVs). Traffic congestion mitigation using real-time information can be enhanced when more and more vehicles are equipped with connectivity technology. There is a growing body of literature that describe promising frameworks using connectivity technology in traffic control, to enhance travel efficiency in urban networks (Abuelenin et al., 2021; al Islam et al., 2021; Dong et al., 2021; Guanetti et al., 2018; Ha et al., 2020; Pupiales et al., 2021; Yang et al., 2021). Also, the deployment of these strategies and technologies has been increasing and have shown to be very cost-effective (U.S Department of Transportation 2012).

1.3. Congestion reduction using vehicle rerouting

ITS-related congestion-mitigation initiatives include vehicle rerouting. This has been found to be particularly useful in dynamic traffic environments (Ho et al., 2019; Li et al., 2009). Vehicle rerouting based on prevailing traffic conditions can improve capacity utilization of the existing roads and ultimately mitigate congestion. For example, car-navigation systems including GoogleMap and TomTom use infrastructure-based traffic information to compute and prescribe traffic-cognizant shortest routes to their users (Wang et al., 2016). Drivers at similar locations, however, may receive similar rerouting guidance. In addition, several large cities have deployed, on a wide scale, traffic guidance systems including Variable Message Signs (VMS), to broadcast real-time traffic flow information. However, as the information is available to all drivers, VMS may provide identical guidance for all vehicles with similar destinations simultaneously. Therefore, these methods tend to merely shift the traffic congestion to other locations of the road network, and the overall congestion issue remains unresolved (Tang et al., 2020).

To address such “congestion-shifting” effects of congestion-mitigation initiatives, multi-route planning algorithms have been proposed in existing literature. One of these is simply to calculate K alternative routes and then randomly assign them to the vehicles (Brennand et al., 2016; Pan et al., 2013). However, such rerouting might yield a further inferior solution because the vehicles that are already close to their destinations may be randomly rerouted by the algorithm to take a longer detour. Therefore, in assigning the routes to the vehicles, it is vital to consider the priority of the vehicles (vehicles’ proximity to intended destination). In addition, it is important to consider the “popularity” or, the frequency-of-use, of each route. As a rule of thumb, vehicles with relatively lower priority should not be assigned routes that are assigned frequently (popular routes). Therefore, from a system efficiency perspective, it is more prudent for lower priority vehicles to be assigned routes with relatively lower popularity. Pan et al. used the Entropy Balanced k Shortest Path (EBkSP) algorithm to dynamically reroute vehicles and demonstrated that the algorithm can efficiently assign vehicles to appropriate routes thus addressing systemwide congestion without shifting, and with reasonably low computational effort (Pan et al., 2013).

1.4. Overview and organization of this part of the report

The remaining parts of the Part I report are organized as follows: Section II presents the underlying concepts of the proposed architecture, and Section III presents the problem settings, which include the framework structure, DRL-stage settings (state space, action space, reward function) and routes assigned stage settings and logical flow. The proposed methodology is introduced in greater detail in the Section IV. Section V, which is the experiment section presents the simulation parameters and introduces the baseline models. Section VI shows the experiment results and analysis from both training stage and testing stage. Lastly, Section VII summarizes the research for part II, offers some concluding remarks, and suggests directions for future work.

CHAPTER 2 UNDERLYING CONCEPTS OF THE PROPOSED FRAMEWORK

In this section, we discuss the prospective role of fog-cloud collaboration in vehicle rerouting, and the attention mechanism for deep reinforcement learning. These are key underlying concepts of the rerouting architecture proposed in this project.

2.1. Prospective role of fog-cloud collaboration in vehicle rerouting

Past researchers have recognized the need for efficient flow of information to support vehicle rerouting. Such a need is critical in urban road networks because such networks are extremely complex and highly interconnected systems where real-time information must be transmitted and disseminated efficiently, otherwise the efficiency and safety of travel could be jeopardized. As a result, information exchange (communication) in a cloud-based computing environment in large networks can be time-consuming (high latency) as the information resources are located in the core. (Aazam et al., 2018). Fortunately, fog information resources, unlike clouds (DataCenters), are located on the edge of the network, and by decreasing the distance from the core to the users, fogs can enhance communication efficiency. Fog nodes refer to distributed fog computing entities that enable the deployment of fog services with processing and sensing capabilities (Marín-Tordera et al., 2017). As shown in the Fig 1, each fog node governs different regions (known as "fog node areas", which are indicated by different colors in the Fig 1). Each fog node collects data (including vehicle speed, vehicle location and vehicle density) in their respective region and preprocesses the data to render it more compact.

Our review of related literature indicates that fog computing has been used to effectively assist in dynamic rerouting. Brennand et al., proposed an ITS architecture using fog nodes they termed "Fog RoutE VEhiculaR (FOREVER)" (Brennand et al., 2017). In that application context, however, the lack of communication among the fog nodes could possibly lead the fogs to recommend routes that are local optimal. Cao et al. designed a traffic congestion scheduling scheme using ITS architecture that incorporates fog computing. In their study, the fog nodes communicated and shared information to characterize overall traffic conditions, and K alternative routes were identified to prevent the same route from always being selected (Cao et al., 2019). In the Cao et al study, even though the fog nodes related to each other, the routes were still calculated locally and therefore the identified routes are most likely locally optimal. Moreover, compared with cloud, fog nodes have relatively weak computing capabilities. In yet another study, Rezaei et al. evoked a fog-cloud based architecture to guarantee that the vehicles are assigned the best routes globally using cloud computing to provide supplementary information where the local information from the fog node is insufficient (Rezaei et al., 2018). They demonstrated that a combination of fog and cloud can represent an efficient architecture that combines local information exchange and global route guidance.

Thus, in this project, the cloud serves as a central platform for planning and making decisions at the system level, while fog nodes are responsible for executing those decisions in a decentralized manner. As shown in Fig 1, the fog nodes collect and preprocess local information and then transfer the preprocessed data to the cloud where system-level decisions are made efficiently. Hence, a centralized-control system with decentralized execution is built on top of a fog-cloud architecture; this arrangement is intended to preserve both the computation capability and the efficiency of information exchange.

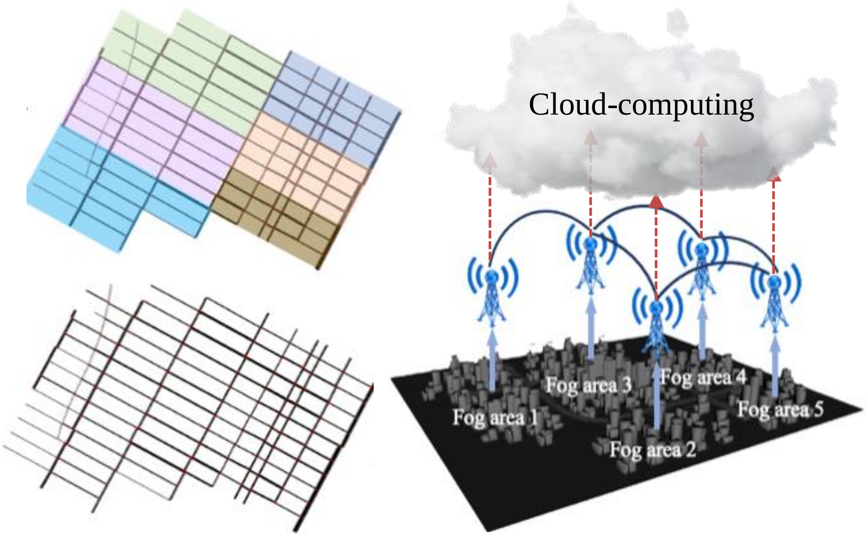


Figure 1 Fog-cloud collaboration

2.2. Attention mechanism for deep reinforcement learning

Given the highly dynamic and complex nature of urban traffic systems, deep reinforcement learning (DRL) can be considered a perfect tool for solving problems such as dynamic rerouting (Zhao et al., 2021). Very few studies in the literature have used reinforcement learning (RL) to address the dynamic rerouting problem. Arkhlo et al. proposed a Multi-Agent Reinforcement Learning (MARL) to identify the best and the shortest path between specified origin-and-destination nodes (Arokhlo et al., 2011). Tang et al. generated an $A*$ trajectory rejection method based on multi-agent reinforcement learning (Tang et al., 2020). Yet still, there exist a few challenges in the use of DRL to address dynamic rerouting problems:

(i). The immense size of the action space, particularly in the case of complex road networks in a large urban area: If all the network edges (links) are considered, then the action space can be as large as ae where a is the number of possible actions and e is the number of edges. This extremely large size of the action space, even for small networks, is costly in terms of training time and inhibits convergence of the algorithm.

(ii). The large size of information collected from the network impairs learning efficiency because not all the information is relevant.

In this project, both challenges are addressed. The fog nodes cover local regions, which represents the edges in these areas Thus, as an alternative to the use of all the network edges in the

RL model, we use the fog nodes, which largely shrink the action space to af where f is the number of fogs. This addresses the first challenge. With regard to the second challenge, the fog nodes' dependency and information flow can be modeled using a graph (Fig 2) where the nodes represent fog nodes (which govern the fog node areas), and the edges represent the connection between fog nodes (fog nodes connect with their neighboring fog nodes). Attention mechanism has been widely used to deal with variable-sized inputs and focus on the most relevant parts of the input to make decisions in graphs (Dong et al., 2020, 2021). Using a Graph Neural Network (GNN) combined with attention mechanisms, Veličković et al created an attention-based architecture to perform node classification of graph-structured data called Graph Attention Networks (GAT) (Velicković et al., 2017). The hidden representations of each node in the graph are calculated by paying attention to the neighbors using a self-attention strategy. Since both local information and neighbor information are crucial for understanding the overall driving environment, a fusion method is needed to explicitly combine such information from different sources. Moreover, there is a need to differentiate the relative importance of input information based on the final decision. Thus, attention mechanism is essential for fogs to automatically "adjust the attention" to relevant information and GAT is an ideal candidate for this attention-fusion task due to its information fusion and attention ability. Therefore, in this paper, GAT is applied in the model to help extract relevant information in the vehicle rerouting (and thus the second challenge is conquered).

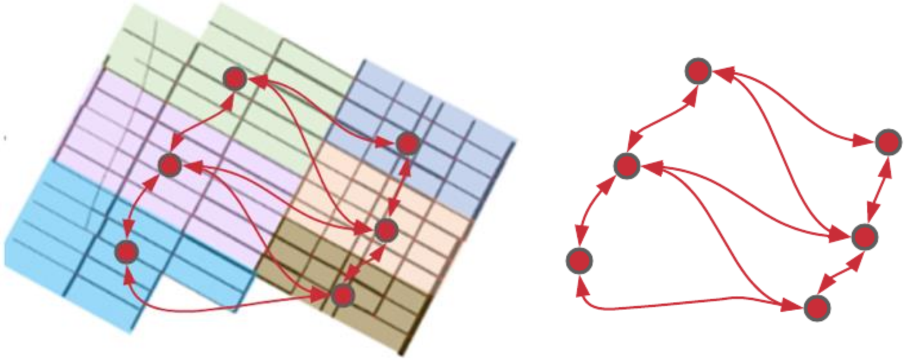


Figure 2 Fog-node graph structure

There are a few research efforts that combined GNN and DRL. Jiang et al proposed a Graph Convolutional Reinforcement Learning (DGN) framework by using GNN as the encoder to learn representations between agents, then have the representations as input to a policy network (Jiang et al., 2018). The joint trading of the encoder and policy network enabled the DGN agents to develop cooperative strategies. Chen et al built a Graphic Convolutional Q Learning (GCQ) framework by combining Graph Convolutional Neural Network (GCN) layer with Deep Q learning for Connected and Autonomous Vehicle (CAV) control. By generating the feature embedded mapping from GCN, and feeding into Deep Q Network, the CAV can make lane-change decisions in a sophisticated manner (Chen et al., 2021). Inspired by this recent research, a DRL model that combines GAT with Deep Q Learning is proposed based on fog-cloud information architecture to extract important and related information to reroute vehicles. The DRL model assists the cloud to make system-level decisions on fog node area road index, which indicates the current and potential congestion level of the specific fog node area.

2.3. Discussion

The DRL-based dynamic rerouting framework proposed in this project can be considered a novel fog-cloud architecture that is carried out in a centralized-planning and decentralized-execution manner. The fog nodes collect and transfer the local information to the cloud. Then a deep learning-based fusion method with graphic attention network is incorporated to generate system-wide decisions considering information from both local and neighboring areas. Then, fog nodes assigned with the decisions help the vehicles to chart their appropriate routes. Moreover, EBkSP method is used to avoid the phenomenon of congestion shifting.

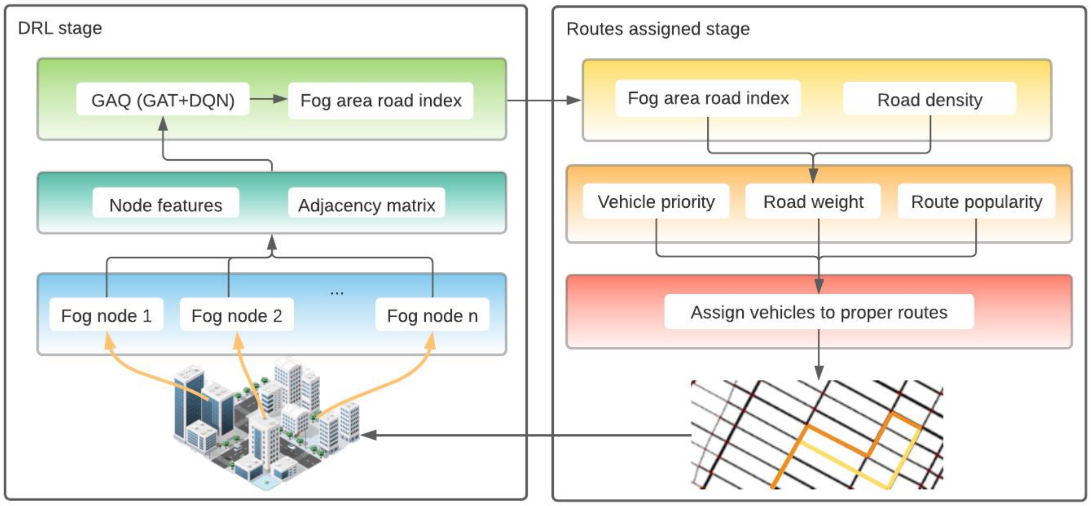


Figure 3 Dynamic vehicle rerouting framework architecture

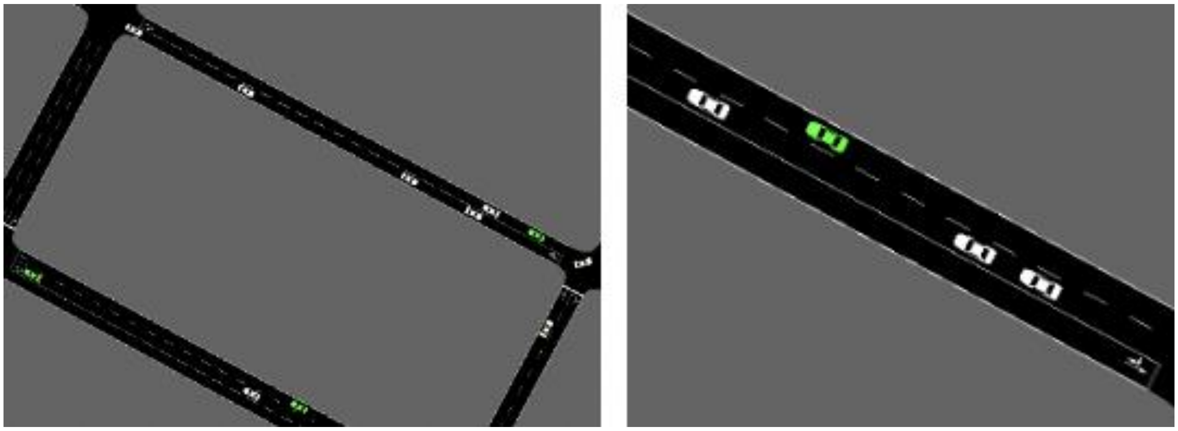


Figure 4 Illustration of BVs (white) and RVs (green) on the road network

CHAPTER 3 PROBLEM SETTINGS

The dynamic rerouting framework consists of two main stages. As shown in Fig 3, in the first stage (DRL stage), the network with fog paradigm is modeled as a graph whose nodes represent different fog nodes and edges represent connections between neighboring fog nodes. The state, action and reward are defined to model the Markov Decision Process in the DRL stage for the agent to make decisions. By applying GAQ, road indexes for different fog node areas (fog node area road index) are generated from the central platform as the control variable for rerouting. In the second stage (route assignment stage), road weights are calculated based on the fog node area road index and road density, then, each vehicle calculates its K alternative shortest paths based on the road weights. Incorporating vehicle priorities and route popularities, the Entropy balance method is applied to assign the appropriate route to each vehicle. After the appropriate routes are assigned, the states of the network are updated, and data are collected to feed into the next episode.

The penetration rate of connectivity technology in vehicles is low even in urban areas [32], [33]. Such lack of widespread connectivity is indicative of existing technology barriers that inhibit prospective dynamic rerouting of a large majority of vehicles in urban road networks. Therefore, this project considers two types of vehicles not only to reduce the number of vehicles considered for rerouting but also to render the study more realistic. Fig 4 indicates rerouting vehicles (RV) (colored green) and Background Vehicles (BV) (colored white). BVs are not rerouted but are incorporated in the framework to add randomness and dynamics in the network. Both vehicles can be detected by fog nodes, and they have distinct origins and destinations.

3.1. DRL stage

At the DRL stage, four key factors are considered: Agent, State space, Action space and Reward function.

Agent: In this research, the cloud represents the agent. At each timestep t, the agent chooses actions $\{a_t^i, i = 1, .., N\}$ for each fog node i based on existing policy and the current environment. After the fog nodes execute the actions, a reward is given to the agent based on the updated states; this motivates the agents to strive for satisfactory results.

State space: The state space includes two parts: node feature at each time step t: X_t; and adjacency matrix at each time step t: A_t. At each t, RVs and BVs can be detected using fog nodes. Thus, network information is extracted from each fog node area. Two types of information are included in the node feature matrix: average speed $\bar{v}_i$ and congestion condition c_i (Maciejewski et al., 2018).

- $\bar{v}_i = \frac{\sum_{k=1,\ldots,n} v_k}{N_i}$ is the average vehicle speed of fog node area i, with N_i equals to the number of vehicles in fog node area i.

- $c_i = \frac{\sum_{G=1,\dots,m} \frac{R^G_{num-veh} \times \tau}{R^G_{num-lane} \times R^G_{len}}}{M_i}$ is the average congestion level of the roads in fog node area i. $\{R^G_{num-veh}, R^G_{num-lane}, R^G_{len}\}$ represent the number of vehicles, length of road G and number of lanes of road G, M_i is the total number of roads in the area i. τ is a scalar to prevent c_i from becoming too small.

The adjacency matrix A_t is a binary matrix with dimension of $N \times N$, where N is the number of fog nodes. A_t reflects the information topology and dependency of the fog nodes. In this study, the graph of the road network is directed, but the graph of the fog layer (for information dissemination purpose) is undirected, and $A_{ij} = 1$ represents the existence of a connection between fog nodes i and j.

Action space: for each time step, each fog node has five different actions to choose $a_i = \{0, 1, 2, 3, 4\}$. The action space for reinforcement learning is aggregated by all possible actions for each fog nodes: $\mathcal{A} = [a_i], i = 1, \dots, n$. The cloud chooses the actions for all the fog nodes and the actions are used as fog node area road index. In the routes assigned stage, the road index is a key factor in the calculation of the road weight which used to generate routes for the rerouting vehicles.

Reward function: In the reward function, we consider both reward and penalty based on average vehicle speed in the network. The purpose of the proposed dynamic rerouting framework is to maintain and enhance the RVs' efficiency. The speed change reflects a change in the traffic conditions. A drastic drop in the average vehicle speed is often symptomatic of congestion. Thus, the framework uses a speed increase reward and speed decrease penalty with threshold of 5 m/s (11 mph).

3.2. Route-assignment stage

The route-assignment stage assigns a route to each rerouting vehicle based on network road weights (calculated by road index from DRL stage and road density) and locations of vehicles. This stage consists of two steps (Fig 5): (a) route computation, (b) route selection. In the route computation step, each rerouting vehicle calculates its K shortest paths based on its current location. Then, in the route selection step, the entropy balanced method is applied to select the appropriate routes (target routes) for rerouting vehicles so as to avoid congestion shifting. The entropy balanced method is based on two critical factors: vehicle priority and route popularity.

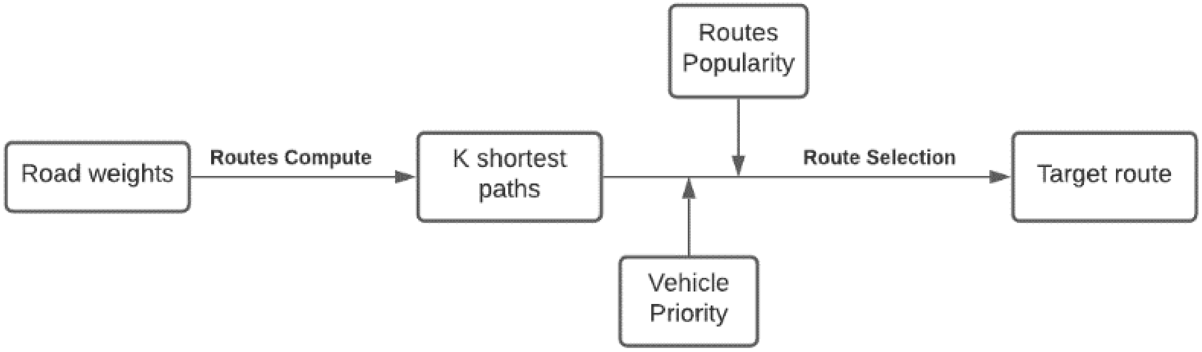


Figure 5 Logical flow of the route assignment stage

Given a set of RVs: $RV = (RV_1, RV_2, \ldots, RV_n)$ to be rerouted, the distance to the destination is used to compute the priority of the RVs. RVs with higher priority can choose the shortest route without considering the popularity of the route, while RVs with lower priority must choose the routes with the lowest popularity to avoid congestion shifting. In this project, two different standards to calculate vehicles' priority are analyzed in the model training stage:

- Priority1-Near: based on the destination of RVs' current location to their destination, RVs that are nearer to their destinations are assigned higher priority.
- Priority2-Far: based on the destination of RVs' current location to their destination, RVs that are further to their destinations are assigned higher priority.

Moreover, the length of the high priority set, which determines the number of high priority RVs is investigated; different lengths of the high priority set are analyzed at the training stage.

CHAPTER 4 METHODOLOGY

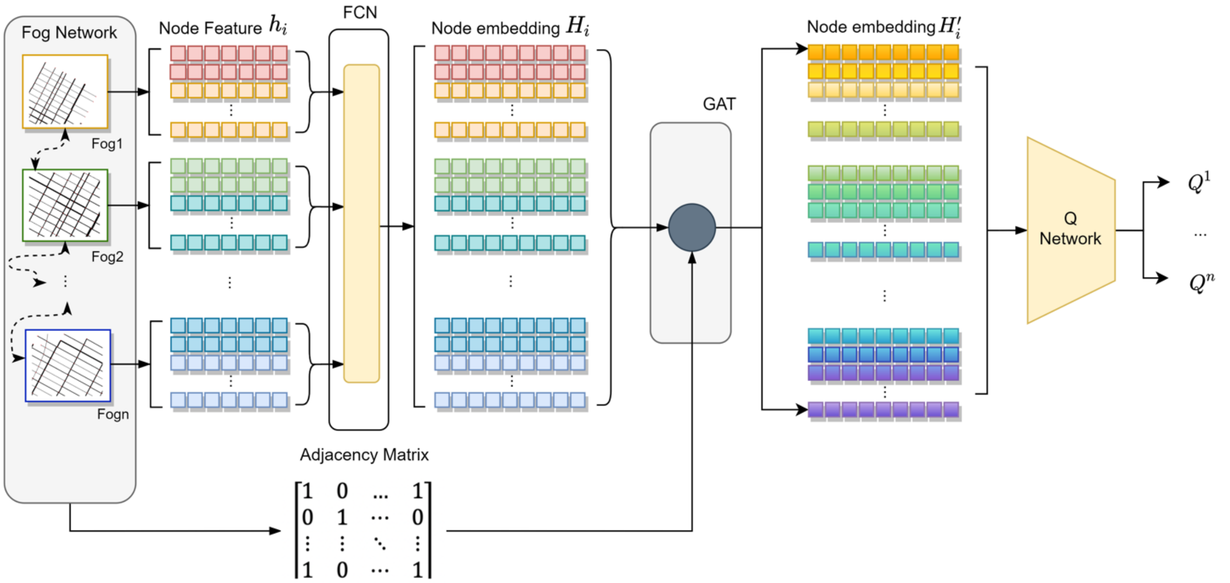
4.1. DRL model architecture

In the DRL stage, the settings are built on top of the fog-cloud architecture with centralized learning but decentralized execution (Chen et al., 2021). Each fog node is assigned with different actions at each timestep, and the target is to improve the efficiency and avoid congestion of the rerouting vehicles in the network. The information attention is modeled with GAT and the decision processor used is Deep Q learning.

At each timestep t, vehicles (RVs and BVs) are detectable by fog nodes. The input of the model is the state s_t. The state is a tuple of $N \times F$ fog nodes feature matrix X_t and $N \times N$ adjacency matrix A_t, N is the number of fog nodes, and F is the number of features in each fog node area. There are two features considered in fog nodes feature matrix: (i) the average speed; and (ii) the congestion condition; fog nodes send their local information to the cloud and then the network node features are concatenated by fog nodes' information. During the information fusion process, the adjacency matrix is used to indicate the spatial relationship between the fog nodes.

As shown in Fig 6, the model consists of the following parts: a fully connected network encoder, a GCN layer, the Q network, and the output layer. At each timestep t, the fog nodes feature matrix X_t is used as the input to the FCN encoder φ to generate node embeddings H_t in d dimensional embedding space

$$H_t = \varphi(X_t) \in \mathcal{H}, \mathcal{H} \subset R^{N \times d} \qquad (1)$$



Fog Network
Fog1
Fog2
Fogn
Node Feature h_i
FCN
Node embedding H_i
GAT
Node embedding H_i'
Q Network
Q^1
...
Q^n
Adjacency Matrix
$\begin{bmatrix} 1 & 0 & \dots & 1 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 1 \end{bmatrix}$

Figure 6 DRL model architecture

Then the graph convolution with attention mechanism is applied to the node embeddings H_t. Unlike the GCN layer, the GAT layer uses the attention mechanism to weight the adjacency matrix instead of using the normalized Laplacian.

$$H_t' = gat(H_t, A_t) = \alpha H_t W + b \tag{2}$$

$\alpha_{ij}$is calculated using the attention mechanism and the adjacency matrix, it represents the coefficient of fog node $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ represents a set of first-order neighbors of fog node i (including i). $T^{\intercal}$ is a weight factor that parameterize the attentional mechanism T:

$$\alpha_{ij} = \frac{exp\left(LeakyReLU(T^{\intercal}[(H_t W)_i \| (H_t W)_j])\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(LeakyReLU(T^{\intercal}[(H_t W)_i \| (H_t W)_k])\right)} \tag{3}$$

The output of GAT layer is the node embedding H_t', which is subsequently sent to a Q network ρ to obtain Q values. Q values are used to evaluate the actions a. With $\hat{Q}$ representing the combined neural network blocks (FCN, GAT, and Q network), ψ representing the combined weights, the model can be expressed as:

$$\hat{Q}_\psi(s_t, a_t) = \rho(H_t', a_t) \tag{4}$$

Experience Replay and Target Network (van Hasselt et al., 2016) are used in the model training to enhance the learning efficiency. Also, the $\hat{Q}$ is trained on randomly sampled batches from replay buffer R with size B to obtain a stable performance. For each batch, the objective is to minimize the value of the loss function:

$$L_\psi = \frac{1}{B} \sum_t y_t - \hat{Q}_\psi(s_t, a_t) \tag{5}$$

Where $y_t = r_t + \gamma \max_a \hat{Q}_\psi(s_{t+1}, a)$.

The architectures of different parts of the network are:

- FCN Encoder φ: Dense (32) + Dense (32)
- GAT layer gat: GATConv (32)
- Q network ρ: Dense (32) + Dense (32) + Dense (64) + Dense (64)
- Output layer: Dense (5)

Warm-up steps are added prior to the training to let the agent explore the environment thoroughly by taking random actions. After the warm-up steps, the training is performed by maximizing the reward and minimizing the losses. Algorithm 1 presents the detailed steps of this process.

Algorithm 1. Graph Attention Q Learning

```
Initialize:
Replay memory $R$
Joint weights $\psi$ and the target network $\hat{Q}_t = \hat{Q}_\psi$
Warm up steps:
For time step $t$ from 1 to $T_w$ do
    • Random actions assigned to each fog node: $a_t = \begin{bmatrix} a^i \\ \vdots \\ a^n \end{bmatrix}$
    • Gather and store $(s_t, a_t, r_t, s_{t+1})$ into the memory buffer $R$
Training steps:
For time step $t$ from $T_w + 1$ to $T$ do
    • Update memory $R$ and choose new batch samples
    • Take $s_t = X_t, A_t$ (Feature Matrix and Adjacency Matrix) and encode the node features
      into a node feature embedding $H_t = \varphi(X_t)$
    • Apply graph attention mechanism $H_t' = gat(H_t, A_t)$
    • Compute Q values for each action combination $a_t$: $\hat{Q}_\psi(s_t, a_t) = \rho(H_t', a_t)$
    • Select optimal action $a_t^* = \underset{a_t}{\operatorname{argmax}} \hat{Q}_\psi(s_t, a_t)$
    • Apply $a_t^*$ to the network and then obtain the reward $r_t$ and next state $s_{t+1}$
    • Add the $(s_t, a_t^*, r_t, s_{t+1})$ into the memory buffer
    • Move from state $s_t$ to state $s_{t+1}$
    • From replay memory buffer $R$, get a random batch size $B$
    • For each training examples with the batch, the target of Q value $y_t$ is calculated:
        ○ If $s_{t+1}$ is not done: $y_t = r_t + \gamma \max_{a_t} \hat{Q}_\psi(s_{t+1}, a_t)$
        ○ If $s_{t+1}$ is not done: $y_t = r_t$
    • Losses are calculated by the loss function: $L_\psi = \frac{1}{B} \sum_t y_t - \hat{Q}_\psi(s_t, a_t)$
    • The target network is updated based on the target updating frequency value
```

4.2. Routes assigned model architecture

In the vehicle route assignment stage, a local search method is applied to assign the proper routes to the RVs. Given the RV set: $RV = \{RV_1, RV_2, \dots, RV_n\}$ At each time step t, after obtaining the fog node area road index, the road weight, which is the actual weight $\left[R_{weight}^{j=1,\dots,M_i}\right]_{i=1,\dots,N}$ for road j in fog node area i can be calculated based on the road index of fog node area i: $\mathcal{R}_{index}^{i}$ and road vehicle density $R_{density}^{j=1,\dots,M_i}$ (number of vehicles):

$$\begin{bmatrix} R_{weight}^{1} \\ \vdots \\ R_{weight}^{M_i} \end{bmatrix}_i = \mathcal{R}_{index}^{i} \times \mathcal{T}_1 I + \mathcal{T}_2 \begin{bmatrix} R_{density}^{1} \\ \vdots \\ R_{density}^{j} \end{bmatrix}_i \tag{6}$$

Where $\mathcal{T}_1$ and $\mathcal{T}_2$ are balance terms to help avoid overwhelming of the road vehicle density on the road index or vice versa. The updated road weights for each road $[R_{weight}^{j=1,\dots,M_i}]_{i=1\dots N}$ are used to calculate K shortest alternative routes for each RV based on their current location. As shown in Equation (7), the K shortest routes set of RV_m ($\{kSP\}_m$) is calculated based on the current location of RV_m ($RV_m^{current}$). The K shortest alternative routes in the $\{kSP\}_m$ set are represented as $r_{s=1,..,k}$.

$$r_{s=1,..,k} \in \{kSP\}_m = ksp(RV_m^{current}) \tag{7}$$

As one of the crucial factors in the routes assigned stage, RVs' priority set $\mathcal{P}$ is obtained by the distance between their current location ($RV_m^{current}$) and the destination, which is represented as d_{RV_m}. According to various priority standards, the RVs' priority set will be sorted differently. In this project, two different priority standards are included. For priority standard 1 (Near), vehicles closer to their destination would have higher priority; for priority standard 2 (Far), vehicles further to their destination would have higher priority.

$$sort_{priority\ standard}(\mathcal{P}) = (d_{RV_m}) \tag{8}$$

Using d_{RV_m}, the priority of the RVs can be determined. The first x vehicles are categorized in the "high priority" set: $\{RV_h\}$ (x can be changed at the stage of model training), the rest of the vehicles are placed in the low priority set: $\{RV_l\}$.

Congestion shifting occurs when vehicles are assigned to the same route. Thus, we need to avoid assigning vehicles to the routes that has already been frequently assigned to vehicles (which is the popular routes). In this project, we solve this problem by incorporating the relative popularity of the routes with the relative priority of the rerouting vehicles' priority in the assignment of routes:

- If RV_m is in high priority set, they will be assigned with the shortest path in their $\{kSP\}_m$ set: $r_s^* = min\{kSP\}_m$.

- If RV_m is in low priority set, the final assigned route is the least popular route from $\{kSP\}_m$ set: $r_s^* = min\{Pop(r_s)\}$. And this not only prevents the congestion shifting but also prevents the RV_m with the final assigned route from an excessively lengthy detour.

The popularity of a route rs, is defined as:

$$Pop(r_s) = e^{E(r_s)} \tag{9}$$

$$E(r_s) = -\sum_{z=1}^{N_{r_s}} \left(\frac{fc_s^z}{N_{r_s}}\right) \ln\left(\frac{fc_s^z}{N_{r_s}}\right) \tag{10}$$

Where N_{r_s} is the number of road segments in the route r_s. fc_s^z $(z = 1, \dots, N_{r_s})$ is the road-weighted footprint of road z in route s, which is calculated from: $fc_z = n_z \times \omega_z$. n_z represents the total number of vehicles assigned to the routes that include road segment z, ω_z is a weight associated with road segment z considers length, lane numbers and average free flow speed:

$$\omega_z = \left(\frac{len_{avg}}{len_z}\right) \times lane_z \times \left(\frac{V_{favg}}{V_{fz}}\right) \tag{11}$$

Algorithm 2 presents the detailed route assignment algorithm.

Algorithm 2. Route assigned by EBkSP

```
Get Roads Weights:
For $RV_m$ in set $RV$ do
    • Find K-alternative shortest path based on current location: $\{kSP\}_m = ksp(RV_m^{current})$
    • Calculate the priority based on current location and add to the set $\mathcal{P} = (d_{RV_m})$
Sorted Priority:
Based on $sorted(\mathcal{P})$, let the top priority RVs into set $RV_h$ and others are low priority RVs into set $RV_l$
Route Popularity:
For $RV_m$ in set $RV_h$ do
    • Assign the shortest route: $r_s^* = min\{kSP\}_m$ to $RV_m$
    • Update the road weight footprint: $fc_s^z, z = 1, \dots, N_{r_s}$
For $RV_m$ in set $RV_l$ do
    • Based on the updated footprint, for the routes in $\{kSP\}_m$, calculate: $Pop(r_s) = e^{E(r_s)}$
    • Assign the least popular route: $r_s^* = min\{Pop(r_s)\}$ to $RV_m$
    • Update the road weight footprint: $fc_s^z, z = 1, \dots, N_{r_s}$
```

CHAPTER 5 EXPERIMENTAL SETTINGS

The proposed framework is implemented in a simulation environment using SUMO (Simulation of Urban Mobility), which is an open-source simulator with well-defined vehicle parameters and vehicle controller (Krajzewicz et al., 2012). The training network is the Manhattan network (Fig 7 (a)) that is imported from OSM (Open Street Map) (OpenStreetMap contributors, 2017) and cleaned in SUMO (Fig 7 (b)), then fog nodes are involved into the network (Fig 7(c)). As shown in Fig 7 (c), there are six fog nodes covers different regions of the network. To have the equivalent information collection of different fog node areas, each fog node covers about 50 roads. Simulator parameters, training parameters and baseline models used in the experiment are discussed in detail in the following sub sections.

5.1. Simulator parameters

In SUMO, critical parameters for the driving simulation environment need to be well defined based on the specific research problem. Detailed description of simulator parameters including network features, scenario parameters, vehicle control parameters, vehicle priority parameters, and training parameters are discussed in the following subsections.

Network features

A $5.926km^2$ area that extracted from the Manhattan area is used in this research, the network includes 287 edges (roads) and 120 nodes (junctions). The network structure is the same as that of the real world. There are multiple road types in the network: 2-lane roads, 3-lane roads, 6-lane roads, and 7-lane roads. Both one-way and two-way roads are included. The speed limit is reflective of the actual real-world conditions as evidenced by data from an open street map. The speed limit varies due to the different road types and ranges from 11 m/s to 28 m/s.

Scenario parameters

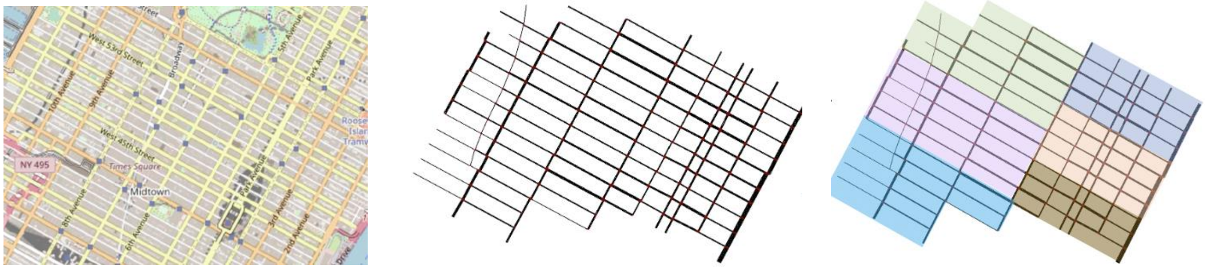
To increase the complexity and to mimic the dynamic nature of the urban road network environment, BVs (colored white) enter the study area from multiple areas with different travel patterns and destinations: (a) from right to the left, (b) from left to the right, (c) from the middle to the top, and (d) from the middle to the bottom; RVs (colored green) enter the map from 3 roads located on the right of the network (two from the top, one from the bottom) and two different destinations are located on the left of the network (one from the middle, the other from the bottom). At the training stage, the inflow rates of the BVs and RVs are both specified as 100 veh/hr. At the testing stage, the inflow rates are changed according to the number of BVs and RVs. A significant factor in the mixed traffic is the penetration rate (which refers the ratio of RVs to the total number of vehicles: $\frac{RV}{BV+RV}$). Therefore, in the training stage, the total number of RVs and BVs is 1000 with a 0.1 RV ratio. While in the testing stage, different RV ratios of the mixed traffic with RVs and BVs are investigated.

Vehicle control parameters: The vehicle control in this project includes vehicle behavior control and routing control. The vehicle behavior control includes car-following control and lane-changing control. In this study, both BVs and RVs use SUMO's built-in car-following and lane-changing controllers. In this research, the routing controller for RVs is based on learning-based model like proposed GAQ-EBkSP model and learning-based baseline model, while the routing controller for BVs is simply based on the shortest path rerouting model. The routing controller runs on different rerouting models; therefore, the performance of different rerouting models can be investigated.

Vehicle priority parameters

As mentioned earlier, there are two ways in which we can calculate vehicle priority (priority1-Near and priority2-Far), both ways are trained in the training stage. Additionally, as discussed in "Model Architecture Assigned to Routes" section of this project, the length of the high priority set is configurable. Thus, different lengths of the high priority set are implemented in the training stage as well.

Training parameters: In the model training section, approximately 800 epochs are trained, with the first 200 epochs as warm-up stage. When training starts, transition batches of size 32 are sampled and put into the model. The optimization parameters used in this research is Adam (Kingma et al., 2015) which has initial learning rate $\gamma = 10^{-4}$.



(a) OSM network (b) SUMO network (c) Network with fog nodes

Figure 7 Network used in experiments

5.2. Baseline models

In this research, the baseline models are the rule-based model and GCQ-EBkSP model:

- Rule-based model: here, the RVs are rerouted using EBkSP only (which means, no learning stage). Rerouting will not be affected by the road index, only the road density (number of vehicles) will be taken into consideration when calculating the road weight.
- GCQ-EBkSP model: there, the RVs are rerouted with learning stage; the road index will be calculated through GCQ model. This model is implemented to compare with the proposed GAQ-EBkSP framework in terms of DRL models' performances.

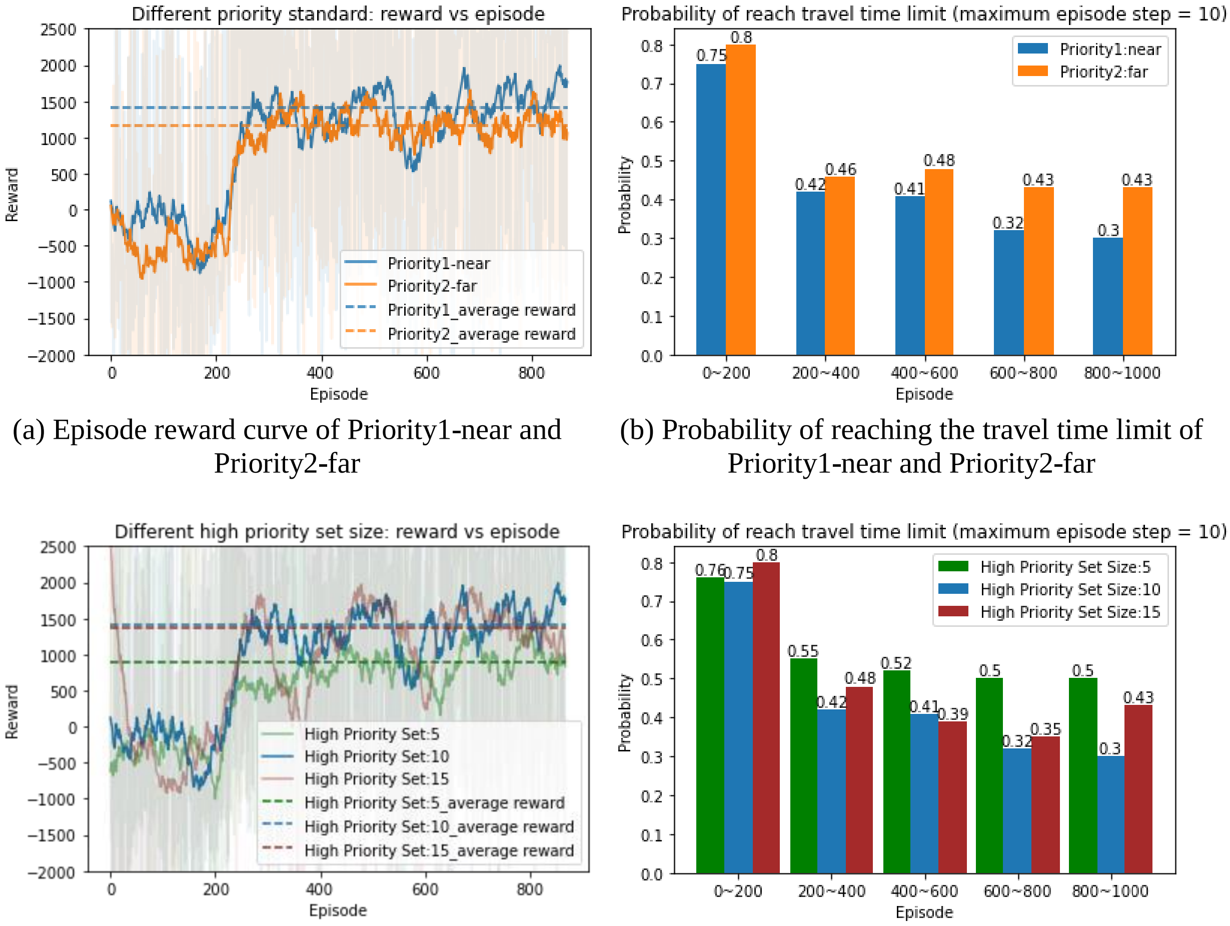
To test the efficacy of learning on the dynamic rerouting of vehicles, the rule-based model is compared with two learning models (i.e., the proposed GAQ-EBKSP and the other baseline model, GCQ-EBKSP). The other baseline model: GCQ-EBkSP provides the means of comparing different DRL model performance under large urban network settings.

CHAPTER 6 RESULTS

6.1. Training stage

At the training stage, the two priority standards (priority1-Near and priority2-Far) are analyzed under the proposed GAQ-EBkSP framework. As shown in the reward curves Fig 8 (a), priority1-Near overperforms priority2-Far. Based on the average reward lines that are calculated after convergence (after 400 episodes) of both priority standards, the average reward of the priority1-Near scenario is higher than that of priority2-Far by approximately 300 units. In this project, the maximum number of steps for one episode is set to be 10 (1 step equals to 200-unit time). Typically, 10 steps per episode is adequate for all the RVs in the network to complete their trips. Therefore, if the maximum number of steps is reached in one episode, it means that rerouting vehicles are unable to complete their trips within the specified maximum number of steps. One evident reason could be that some of the RVs encounter severe congestion with BVs. Fig 8 (b) presents the probability of reaching the maximum episode steps of different priority standards throughout the training period. Clearly, as the training progresses, both priority standards show some progress, the probability of reaching the maximum episode step is lower. This means that the GAQ-EBkSP framework effectively prevents the rerouting vehicle from encountering severe congestion. On the other hand, using priority1-Near scenario, there is 13% lower probability (compared to priority2-Far) to encounter severe congestion.

With the exception of different priority standards, the length of the high priority set, which determines the number of high priority RVs is also crucial factor for the training stage. Therefore, we implemented three different set lengths for the high priority case: {5,10,15} under the priority1-Near standard. As shown in Fig 8 (c) and Table 1, on the basis of the average reward lines (calculated after convergence), the performance of high priority length 5 (average reward of 893) is much worse compared to that of length 10 (which indicated an average reward of 1420) and that of length 15 (average reward of 1371). Even though the average rewards of high priority set length 10 and 15 are close, 10 is still a superior choice. This is because when the high priority set length is relatively large, almost all the vehicles in the network choose the shortest route without considering route popularity, which leads to congestion shifting from one part of the road network to another. As shown in Fig 8 (d), the probability of getting severe congestion using 15 as the high priority set length is higher than that associated with a set length of 10. Throughout the training process, the combination of priority1-Near as the priority standard and 10 as the high priority set length overperforms other combinations. Therefore, this combination is used in the proposed model.



(a) Episode reward curve of Priority1-near and Priority2-far

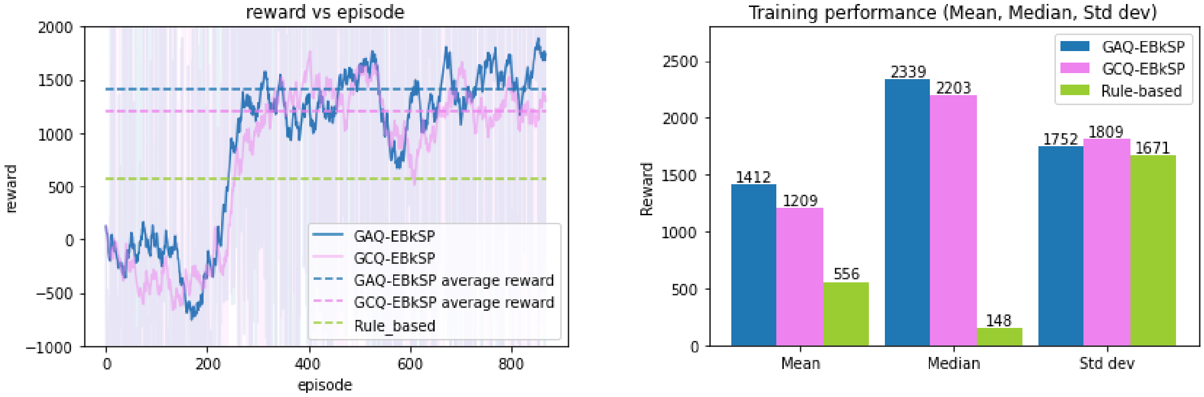
(b) Probability of reaching the travel time limit of Priority1-near and Priority2-far

(c) Episode reward curve for different sizes of the high priority set

(d) Probability of reaching the travel time limit, for different sizes of the high priority set

Figure 8 Training performance of different priority standards

Fig 9 (a) presents the episode reward curve of rule-based model, GCQ-EBkSP and GAQ-EBkSP (proposed RL model). The proposed GAQ-EBkSP and baseline GCQ-EBkSP both outperform the rule-based model. Since the rule-based model equips with no learning stage, it can be difficult to obtain improvement (in terms of reward), and the average reward of the proposed model is approximately 850 units higher than the rule-based model (As shown in Fig 9 (b) and Table 2). By using the GAQ model's attention mechanism as a replacement for the statically normalized convolution operation used by the GCQ model, the GAQ model obtains superior learning efficiency through the consideration of the importance of adjacent information. Therefore, the GAQ model obtains higher learning efficiency by providing road index considered different importance of neighboring information. As shown in Fig 9 (a) and (b), the proposed GAQ-EBkSP model performs superior to the baseline GCQ-EBkSP model (by approximately 17% additional reward units).



(a) Reward comparison for each episode

(b) Training performance comparison (Mean, Median, Std dev)

Figure 9 Reward comparison (Proposed vs. Balanced)

Table 1 Training performance comparison (episode reward) with different parameters

Parameter	Statistic	Training Scenario
Priority standards		
Priority1-Near	Mean	**1412**
	Median	**2339**
	Std dev.	1752
Priority2-Far	Mean	1172
	Median	1855
	Std dev.	**1667**
High priority set length		
High priority set: 5	Mean	893
	Median	755
	Std dev.	1649
High priority set: 10	Mean	**1412**
	Median	**2339**
	Std dev.	1752
High priority set: 15	Mean	1371
	Median	2262
	Std dev.	**1751**

Table 2 Training performance comparison (episode reward) for each model

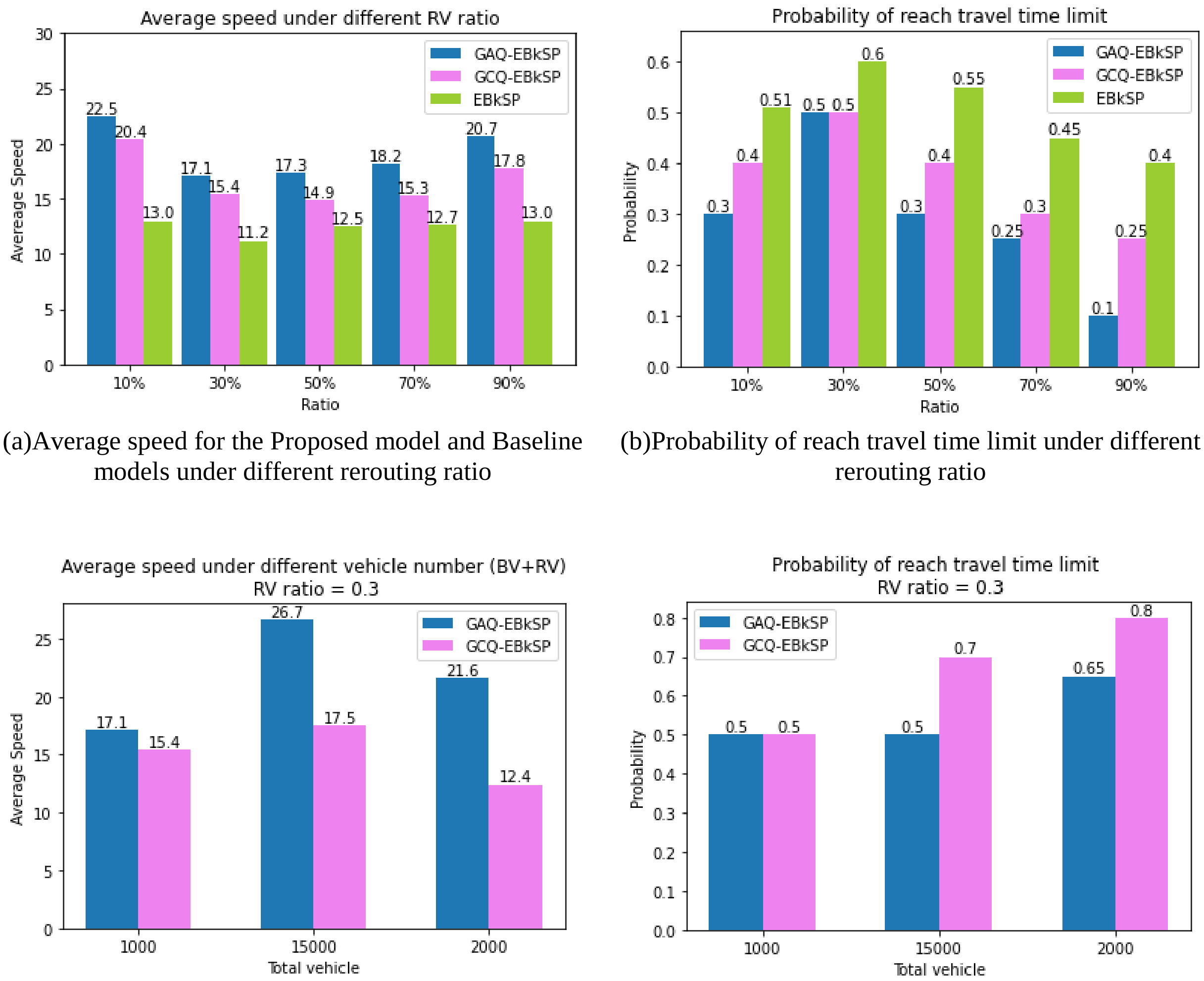
Model	Statistic	Training Scenario
Graph Attention Q network (GAQ)	Mean	**1412**
	Median	**2339**
	Std dev.	1752
Graph Convolution Q network	Mean	1209
	Median	2203
	Std dev.	1809
Rule-based	Mean	556
	Median	148
	Std dev.	**1671**

6.2. Testing stage

Two important factors are considered in the testing stage: (i). the RV ratio (ii). the total number of vehicles in the network (BV + RV). For the RV ratio test, the total number of the vehicles is set to be 1000. Five different scenarios with RV ratios range from 0.1~0.9 were tested: 0.1 (900 BVs and 100 RVs), 0.3 (700 BVs and 300 RVs), 0.5 (500 BVs and 500 RVs), 0.7 (300 BVs and 700 RVs), 0.9 (100 BVs and 900 RVs). Higher ratio reflects larger number of RVs in the network per unit time. Thus, the inflow parameter (vehicles per hour) is adjusted to increase with the RV ratio to maintain RVs' number in the network under different ratios. For the total number of vehicles test, three scenarios with different total numbers of vehicles with fixed RV ratio r_{RV} are generated: 1000 ($1000(1-r_{RV})$ BV and $1000r_{RV}$ RV), 1500 ($1500(1-r_{RV})$ BV and $1500r_{RV}$ RV), 2000 ($2000(1-r_{RV})$ BV and $2000r_{RV}$ RV). The performance metrics are the average speed and the probability to encounter severe congestion, which reflect the efficiency of the proposed method under the different scenarios.

As shown in Fig 10 (a) and Fig 10 (b), the proposed GAQ-EBkSP model performs the best among all the models under different ratios in both average speed and probability of reaching travel time limit (probability of the RVs encountering severe congestion). Interestingly, the learning-based models (GAQ-EBkSP, GCQ-EBkSP) outperform the rule-based model across all scenarios considered. Fig 10 (a) represents the average speed of RVs, the rule-based model reroutes the vehicles based on the current density of the network only with no learning stage to foresee the potential congestion at different road sections. As a result, the vehicles have no "future planning" and cannot choose superior routes jointly. This is the main reason why the learning-based models achieve higher reward values. Particularly, this is observed where the RV ratio is lower (a low ratio means that there are more background vehicles that are not under rerouting control), it is more likely to encounter severe congestion. As shown in Fig 10 (b), when the rerouting ratios are relatively low (≤30%), the average probability of encountering severe congestion is 0.35 when the learning-based model is used, while the rule-based model has an average probability of 0.51 which is 21% higher than that of the learning-based model. When the rerouting ratios increases, the probabilities of encountering severe congestion are lower when either model is used. Yet still, even in the case, the learning-based model outperforms the rule-based model. Furthermore, the GAT layer in the proposed GAQ-EBkSP model expands the basic aggregation function of the GCN layer in the GCQ-EBkSP model, assigning different importance to each edge through the attention coefficients. Thus, compared to the GCN layer, GAT layer are able to learn the information which is much more relevant to the problem. As shown in Fig 10 (a) and (b), the proposed GAQ-EBkSP model outperforms the GCQ-EBkSP model in all the scenarios considered.

Based on Fig 10 (a) and Fig 10 (b), the worst case is when the RV ratio is 0.3. Thus, we use 0.3 as the fixed RV ratio in the test that investigates the effect of a different total number of vehicles on the two learning-based models. As shown in Fig 10 (c) and Fig 10 (d), the proposed model has superior performance with regard to both the average speed and probability of RVs encountering severe congestion. As indicated in Fig 10 (c), the proposed GAQ-EBkSP exhibits a higher level of robustness compared to the baseline GCQ-EBkSP. When the total number of vehicles is 1500, the average speed of the RVs under the proposed model is still high, indeed, higher than the scenario with 1,000 vehicles. This because as the number of RVs increases, some of the fast-moving RVs overtake the slow-moving RVs. In the Fig 10 (d), the probability of achieving the travel time limit is 0.5, which is the same with the scenario that has 1000 vehicles. Even with 2,000 vehicles (i.e., a very high chance of congestion), the average speed is still promising. The proposed GAQ-EBkSP model outperforms the GCQ-EBkSP baseline model, in all the scenarios considered. The benefits of using the attention mechanism in the proposed model are shown evidently when there is massive information (the total number of vehicles is large). As shown in Fig 10 (c) and (d), when the total number of vehicles reaches 1500 and 2000, the average speed of RVs in the proposed model is nearly 10m/s higher than that of the RVs in the baseline model. Moreover, when using the baseline model, the probability of RVs encountering severe congestion under large total number (1500 and 2000) of vehicles is 75%, while this probability is only 57.5% when the proposed model is used.



(a)Average speed for the Proposed model and Baseline models under different rerouting ratio

(b)Probability of reach travel time limit under different rerouting ratio

(c)Average speed for the Proposed model and Baseline models under different total number of vehicles

(d)Probability of reach travel time limit under different total number of vehicles

Figure 10 Testing performance: different scenarios for rule-based and RL-based models

CHAPTER 7 CONCLUDING REMARKS

In this part of the project, a DRL (GAQ)-EBkSP model based-on fog-cloud architecture is proposed to dynamically reroute the vehicles in large transportation networks. The setting of the proposed model follows a centralized learning and decentralized execution manner. The fog nodes collect regional information and send to the cloud, where the road indexes of different fog node areas are learned. After obtaining the fog node area road indexes, EBkSP method is used in cloud to search the proper routes for the rerouting vehicles in their K shortest routes set based on the vehicles' priority and routes' popularity levels. Moreover, the large action space problem in large transportation network is solved by using fog nodes to substitute regional edges. Furthermore, the project applied a graph attention mechanism to fuse information and extract relevant information to enlarge the learning efficiency. The cloud layer helped ensure that the assigned routes are not local optimal but global optimal, and the routes are assigned to the vehicles based on their priority and routes' popularity to avoid the congestion shifting. A region in mid-Manhattan, New York, is used as the experiment network study area. Different levels of the RV ratios (0.1~0.9) and total numbers of vehicles (1000, 1500, 2000), are tested. The testing results suggest that the proposed model (GAQ-EBkSP) outperforms the baseline models (rule-based model and GCQ-EBkSP) in terms of average speed and the probability of reaching the travel time limit in various scenarios; the learning-based model (proposed GAQ-EBkSP, GCQ-EBkSP) outperform the non-learning-based model (rule-based model) across different scenarios.

The fog nodes layer plays a crucial rule in the developed framework. In this research, six fog nodes are used to cover the network. However, different numbers of fog nodes and different fog node area sizes are expected to influence the rerouting decision. Thus, in the future work, the impacts of different number of fog nodes and different fog node sizes can be studied.

PART II

System Control using Fog-Cloud Based Multiagent Reinforcement Learning and a Case Study involving Scalable Traffic Signals

CHAPTER 8: INTRODUCTION

With growing global populations, increased urbanization, and trends of growing automobile ownership, urban transportation networks are increasingly subjected to traffic congestion. The consequential loss of time, increased emissions, and reduced safety in urban transportation can be expected to grow along with increased congestion. The optimization and control of traffic signals represent a key strategy for the management of traffic congestion and improving traffic conditions in urban areas. According to the Federal Highway Administration (FHWA), poor signal timing can account for up to 10% of traffic congestion (FHWA, 2020). Further, implementation of advanced traffic signal control (TSC) systems in Phoenix, Arizona has seen reductions in traffic collisions by 6.7%, travel times by 11.4%, and delay by 24.9% (Zhao et al., 2011). Therefore, developing and deploying advanced TSC systems can be integral to improving urban traffic conditions.

Traffic signal control is a domain that has seen much attention and research due to its direct impact on social and commercial activities. Broadly, TSC can be classified into two categories: fixed-time traffic control and real-time traffic control. Fixed-time traffic control typically uses a pretimed program that controls the cycle and split times. Webster (1958) was one of the earliest researchers to present a fixed-time control model, which aimed to minimize average delay of vehicles (Webster, 1958). For traffic flow conditions that are stable and do not exhibit randomness, fixed-time traffic control is well-suited. However, for traffic flow conditions that exhibit high levels of stochasticity and instability, fixed-time traffic control models are unsuitable due to their static nature. A better alternative are real-time traffic control models that are responsive to traffic conditions. Traffic control strategies can be made using real-time traffic data, allowing signals to adjust accordingly with unstable and/or stochastic traffic flow. A widely used real-time traffic controller is the actuated signal, which regulate its cycle and timings according to the detector and sensor inputs of the real-time traffic. While many applications of actuated signals have been developed and deployed to great effect, they suffer from the inability to cooperate with many other intersections and do not utilize queues of other phases. Therefore, actuated traffic controllers are unsuitable for addressing network-wide control of urban intersections.

In most urban areas, travel patterns are highly dynamic, and traffic signals are deeply interconnected. Poorly designed signal timings can paradoxically exacerbate congestion, especially when a locally optimal solution is scaled up to large networks. While several studies have shown that traffic signal control (TSC) methods such as actuated and pretimed controls are adequate for small networks (Koonce and Rodegerdts, 2008; Ceyland and Bell, 2004), they cannot be integrated effectively into large networks. With the imminent emergence of robust vehicular connectivity and automation technologies, many solutions on traffic signal control leveraging such technologies are being studied. Guo et al., (2019) presented six types of connected and automated vehicle- (CAV) based traffic control methods including an improved actuated system that utilizes CAV data (Guo et al., 2019). However, the question of when the CAVs will be deployed into the real-world is still largely debated. As such, this study aims to provide an intelligent, scalable traffic control model that can be integrated into large, urban networks without utilizing CAVs directly.

The prospect of scaling small-intersection TSC solutions to larger networks has been a persistent challenge that has been addressed using a variety of optimization algorithms. In recent years, there has been pronounced interest in the investigation of other solutions methods, and this new direction is motivated by advancements not only in computer hardware and software, including computing power, but also in techniques and technologies for data management and analytics including artificial intelligence and machine learning. For example, multi-threaded, multi-core central processing units (CPUs) such as the Ryzen Threadripper series with up to 64-cores and 128-threads have become more widely available for consumer use. Combined with advances in graphics processing units (GPUs) and large video random access memory (vRAM) capabilities, training deep reinforcement learning models has become much more efficient in recent years. It is acknowledged that deep learning and reinforcement learning concepts were introduced several decades ago (Jin, 1992; Tesauro, 1995). However, recent advancements in computational capabilities have made their application more feasible and therefore have fostered a new generation of deep and reinforcement learning algorithms in continuous and discrete control (Lillicrap et al., 2016; Tesauro, 1995). Alongside emerging smart infrastructural technologies that facilitate real-time data collection and sharing such as road-side units (RSUs) and drones, the implementation and deployment of scalable TSCs and other intelligent transportation systems have become increasingly feasible.

For these reasons, deep reinforcement learning (DRL) based approaches to solving TSC problems in large networks has become an increasingly studied topic. Wiering's study was one of the earliest to propose the use of reinforcement learning algorithms for traffic signal control to minimize city-wide congestion (Wiering et al., 2000). Prashanth and Bhatnagar proposed reinforcement learning with function approximation for traffic signal control, using Q-learning for adaptive signal control (Prashanth and Bhatnagar, 2010). Chu et al proposed a multiagent deep reinforcement learning algorithm that could be applied to large-scale networks; they applied an actor critic network to recurrent neural network with long-short term memory (LSTM) (Chu et al., 2020). Wang et al, proposed the cooperative double Q-learning (Co-DQL) model that leverages mean field approximation of all other agents in the network to significantly reduce model complexity and the curse of dimensionality (Wang et al., 2021).

While the aforementioned studies utilize the state-of-the-art DRL approaches for TSC problems, an oft overlooked topic is the resource constraints that may restrict transportation agencies and other government entities from deploying data-facilitating infrastructure such as RSUs and drones. As such, this study presents an alternative perspective to scalable TSC models that can reduce the number of deployed data-facilitating infrastructure. In essence, the proposed model utilizes a graph attention network (GAT) to preserve the topology of the traffic network while focusing on relevant inputs to make decisions. Doing so allows the model to address large networks as well as variable sized inputs. RSUs are deployed in an urban grid-like network, each serving as fog-nodes that collect data via detectors and share with other fog-nodes in its range, utilizing the information to control the phase and duration of the traffic lights in its control. The Q-network utilizes double estimators to approximate $\max_{a} E\{Q_t\ (s_{t+1}, a)\}$ instead of maximizing over the estimated action values in the corresponding state to approximate the value of the next state (as is the case in standard Q-learning), performance overestimation is avoided. Overall, the model extracts node embeddings from fog node features while also constructing an adjacency matrix that maps the topology of the connected fog nodes, which are passed through the attention layer to be used for the Q-network. To the best of the authors' knowledge, this is the first study that considers preservation of network topology in TSC problems through the use of GATs.

CHAPTER 9. BACKGROUND FOR THE METHODOLOGY

9.1 Reinforcement Learning

Figure 11 presents the architecture for the GAT Model. In general, reinforcement learning (RL) utilizes feedback of decisions, observations, and rewards. Deep reinforcement learning (DRL) combines RL with deep learning, which allows for end-to-end training of multilayer models that can solve complex problems. This is particularly useful for sequential decision making such as in robotics, video games, and traffic operations (Lillicrap et al., 2016; Chu et al., 2020; Vinitsky, et al. 2018; Ha, et al., 2020; Liu and Yang, 2019.

One of the most popular single-agent RL method is Q-learning. Q-learning is a model-free reinforcement learning approach that can be considered as asynchronous dynamic programming, where agents learn optimal policies in Markovian domains through solving sequential decision-making problems (Watkins, et al. 1992). This is achieved through estimating the optimal value, $Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$, for each action a during state s. Because most problems have large state and action spaces to learn all action values separately, a parametrized value function $Q(s, a; \theta_{\square})$ can be learned instead. Thus, the standard Q-learning update for the parameter from taking action $a_{\square}$ in state $s_{\square}$ with observed reward r_{t+1} and the subsequently resulting state $s_{\square}$ is:

$$\theta_{t+1} = \theta_t + \alpha \left(Y_t^Q - Q(s_t, a_t; \theta_t) \right) \nabla_{\theta_t} Q(s_t, a_t; \theta_t)$$

where α is the learning rate, and the target Y_t^Q is defined as:

$$Y_t^Q \equiv r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta_t)$$

where the constant $\gamma \in [0,1)$ is the discount factor adjusting the weight between immediate and later rewards.

Q-learning in multiagent reinforcement learning (MARL) differs primarily in that MARL is based on Markov game instead of a Markov decision process (MDP) (Shapley, 1953; Watkins, et al., 1992). Similarly to MDPs, Markov games can be represented as a tuple $(M, \boldsymbol{S}, \boldsymbol{A}_{1,2,\ldots,M}, r_{1,2,\ldots,M}, p)$, where M is the number of agents, $\boldsymbol{S} = \{s_1, s_2, \ldots, s_m\}$ is the set of system states, $\boldsymbol{A}_m$ is the action set of agent $m \in \{1,2, \ldots, M\}$, $r_m: \boldsymbol{S} \times \boldsymbol{A_1} \times \ldots \times \boldsymbol{A_M} \times \boldsymbol{S} \rightarrow \mathbb{R}$ is the reward function for agent m, and $p: \boldsymbol{S} \times \boldsymbol{A_1} \times \ldots \times \boldsymbol{A_N} \rightarrow \mu(\boldsymbol{S})$ is the transition function for moving from one state s to another state s' given action $a_{1,2,\ldots M}$. Partially observable Markov games additionally require Ω, the set of observations of the hidden states, and $\mathcal{O}: \boldsymbol{S} \times \Omega \rightarrow \mathbb{R}_{\geq 0}$, the observation probability distribution.

In MARL, each agent learns to choose its actions according to their respective strategies. At each time step, the system state transfer occurs by taking the joint action $a = (a_1, \ldots, a_M)$ under the joint strategy $\pi \triangleq (\pi_1, \ldots, \pi_M)$, and each agent receives their immediate reward from the joint action. For each agent m under joint policy π and initial state $s(0) = s \in \boldsymbol{S}$, the expected discounted reward is:

$$V_m^{\pi}(s) = E^{\pi}\left\{\sum_{t=0}^{\infty} \gamma^t r_m(t+1) | s(0) = s\right\}$$

Additionally, the agent-specific average reward can be found as:

$$J_m^{\pi}(s) = \lim_{T\to\infty} \frac{1}{T} E^{\pi}\left\{\sum_{t=0}^{T} r_m(t+1) | s(0) = s\right\}$$

9.2 Graph Neural Networks

Graph neural networks are able to preserve acyclic and nonacyclic graph topology, which can enhance road network representation particularly in the context of scalable network traffic signal control (Watkins, 1989; Wang, et al. 2020; Wei, et al. 2019). Deep reinforcement learning requires a strong neural network architecture for forward and backpropagation for model training (Devailly et al., 2021). Graph convolutional networks (GCNs) can serve as powerful neural networks that can address graph data for deep reinforcement learning (Goodfellow, et al. 2016; Kipf and Welling, 2017). The nodes of a GCN layer aggregates its own observed states and those of its neighbors into embeddings. Given different relational graphs, the message propagation is as follows (Kipf and Welling, 2017):

$$h_{\square}^{l+1} = \varsigma\left(\Sigma_{m\in\mathscr{M}_i} g_m\left(h_i^l, h_j^l\right)\right)$$

where $h_{\square}^{l} \in \mathbb{R}^{d^{(l)}}$ denotes the hidden state of node $v_{\square}$ in the l^{th} layer of the neural network, $d^{(l)}$ is layer dimensions, $\mathscr{M}_{\square}$ is the set of incoming messages, and $g_{\square}(\cdot)$ is the transformation for the message from the nodes.

In essence, these node embeddings can address problems caused by variable length inputs to perform various sequential learning tasks given graph data, and error terms can be used to backpropagate to perform the requisite gradient descent for parameter tuning purposes.

CHAPTER 10. METHODOLOGY

10.1 DRL Model Architecture

The fog-based graphic RL (FG-RL) model for TSC presented in this paper employs a scalable and decentralized methodology. The graphical structure of the network topology is preserved with traffic signals and intersections, along with their relative adjacencies. The fog arrangement determines the topology of the connected entities and the number of the connected intersections within its range. Therefore, the adjacency matrix containing the relative adjacencies and connectivity of intersections vary corresponding to how the RSUs (and in turn, the fog nodes) are dispersed in the network and how many intersections each RSU oversees. In DRL architecture, each RSU is represented as a fog node, which serves as an agent that makes decisions to select traffic signal phases for each of the intersections it oversees, with an overall goal to reduce congestion.

The network topology and information attention are modeled using GAT. Figure 11 presents the network architecture.

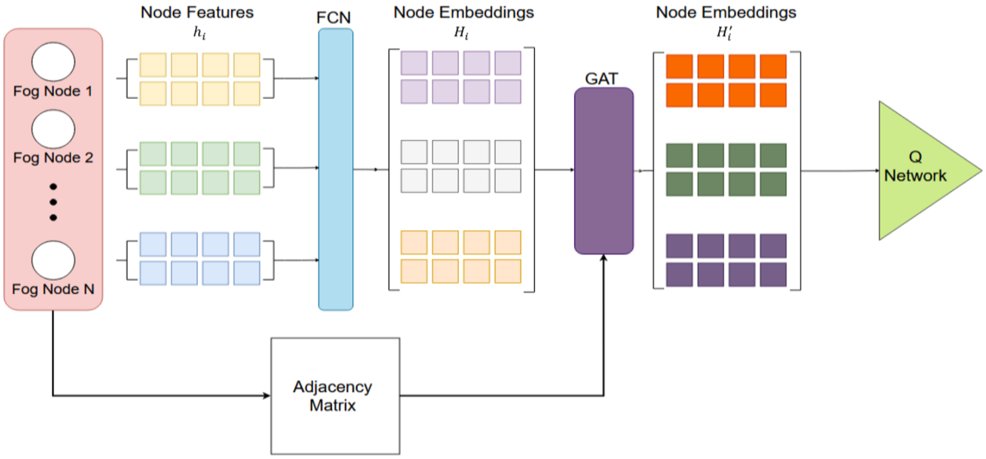


Figure 11: GAT Model Architecture

The fog node can oversee multiple intersections, some of which may have few or no queued vehicles. Therefore, it must learn to divert attention away from relatively uncongested intersections and focus more on congested intersections. However, a given intersection's congestion levels can vary drastically between episodes or even across different time-steps in one episode. As a result, applying an attention model can facilitate the learning process under conditions when such variations exist.

Each fog node i produces node embeddings that encode node features h_i. The state is a tuple of $N \times F$ node feature matrix X_t and an $N \times N$ adjacency matrix A_t, where N is the total number of nodes, and F is the number of features in each node. The feature matrix considers the states consistent with those in the literature (Wang, et al. 2021; Chu, et al., 2020), namely, (i) the cumulative delay of the first vehicle in each incoming lane at an intersection, and (ii) the total number of approaching vehicles in each incoming lane.

At each time-step t, the node feature matrix X_t is fed as the input into a fully connected encoder denoted φ that generates node embeddings H_t in d dimensional embedding space $\mathcal{H} \in \mathbb{R}^{N\times d}$

$$H_t = \varphi(X_t) \in \mathcal{H}$$

The node embeddings then are passed through the graph convolution with attention mechanism.

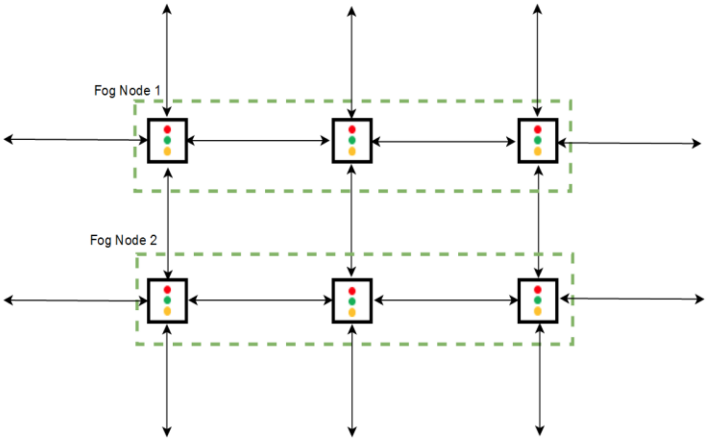


Figure 12: Small TSC Network

The adjacency matrix is weighted using the attention mechanism:

$$H_t' = GAT(H_t, A_t) = \alpha H_t W + b$$

where α_{ij} are coefficients computed by the attention mechanism defined in the literature (Velickovic et al., 2017):

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(\boldsymbol{a}^T[\boldsymbol{W}h_i||\boldsymbol{W}h_j]\right)\right)}{\Sigma_{k\in(\mathcal{N}_i)}\exp\left(LeakyReLU(a^T[\boldsymbol{W}h_i||\boldsymbol{W}h_k])\right)}$$

The output of the GAT layer is then used as inputs to the Q network to obtain the Q values. Further, experience relay and soft target update are utilized to enhance learning (Mnih et al., 2013; Hester et al., 2018), and the model is trained on randomly sampled batches from a replay buffer. Thus, the architecture can be summarized as follows:

- FCN Encoder φ: Dense (32) + Dense (32)
- GAT Layer GAT: GATConv (32)
- Q Network: Dense (32) + Dense (32) + Dense (64) + Dense (32)
- Output Layer: Dense (5)

CHAPTER 11. CASE STUDY FOR PART II

The case study utilized the Simulation of Urban MObility (SUMO) for traffic simulation (Krajzewicz et al., 2012), an open-source simulator that enables detailed tracking of vehicle and traffic light parameters. For an initial proof of concept, a small 6-node network is considered (Figure 2).

11.1 Network Descriptions

A small grid network was used for numerical experimentation, as shown in Figure 2. The first setting utilizes a "smart cities" approach, where each intersection is connected via a central controller in a cloud environment. This setting is a fully observable MDP. It must be noted that this is an ideal setting that has no constraints, meaning that all entities are assumed to be connected. While this can be achieved easily in simulation, it will need many connectivity facilitating infrastructure units to ensure that the entire network is connected. Especially in large networks, this can be problematic.

The second setting utilizes the proposed fog-node approach, where intersections are grouped together by a small number of connectivity facilitating infrastructure such as RSUs or drones. Specifically, for this numerical example, two fog nodes are deployed such that the upper horizontal intersections are connected, and the lower horizontal intersections are connected. As previously states, the two benefits of segmenting the whole network into smaller fog nodes is the improved scalability and the possibility of reducing the number of RSUs/drones required to facilitate the intelligent TSC models.

Each westbound and eastbound road segment entering signalized intersections is a two-lane arterial comprised of a through-lane and a left-turn lane. Each northbound and southbound road segment consists of a single through-lane. Vehicles enter each outer road segments (10 total) at a flow rate of 2200 vehicles/hour. The vehicle origins and flows are randomly distributed.

11.2 MDP Settings

Action space:

Each fog node controls the three traffic signals in its range. As shown in Figure 2, Node 1 controls the top three signals, and Node 2 controls the bottom three signals. Each signal can take one of five pre-determined phases, as is consistent with most literature and the practice (Wang, et al. 2021; Chu, et al., 2020): east-west straight, east-west left-turn, three straight and left-turn phases for east, west, and north-south.

State space:

The local state observed within each fog node is defined as follows:

$$s_{k,t} = \{wait_{k,t}[lane], wave_{k,t}[lane]\}$$

As stated previously, $wait_{k,t}[lane]$ denotes the cumulative delay of the first vehicle for a given lane in an intersection, and $wave_{k,t}[lane]$ denotes the total number of approaching vehicles along each incoming lane.

Rewards:

The reward function consists of two main penalties:

$$r_1 = wait_{k,t}[lane]$$
$$r_2 = wave_{k,t}[lane]$$

The total reward is the negative weighted sum of the two penalties,

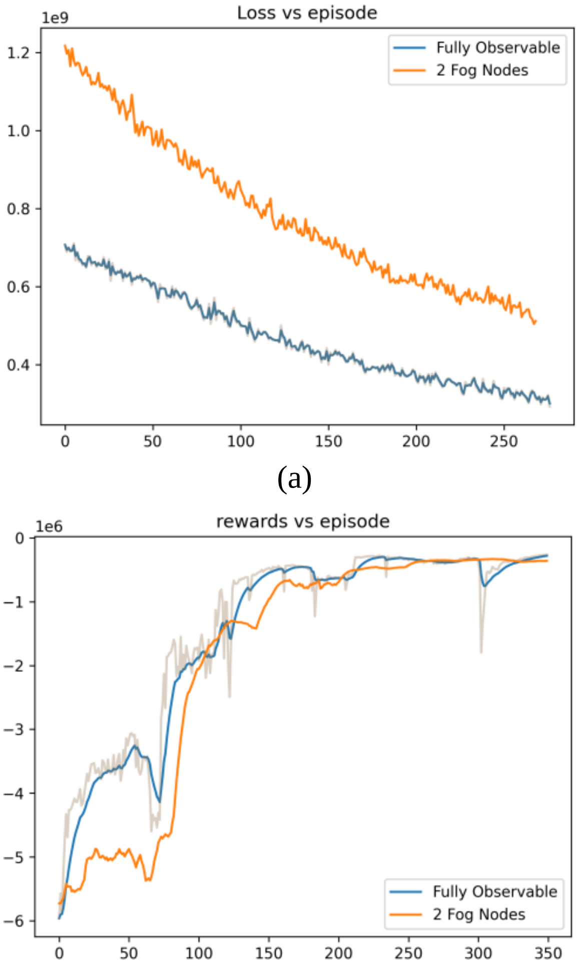
$$r = -\Sigma_{lane}(\sigma_1 r_1 + \sigma_2 r_2)$$

where σ_1, σ_2 are used to scale the two penalties.

This numerical example used $\sigma_1 = 1$ and $\sigma_2 = 0.30$.

11.3 Preliminary Results

Fig. 13 presents a comparison of training results using 2 fog nodes vs. a fully observable system.



(a)

(b)

Figure 13: Comparison of Training Results using 2 Fog Nodes vs Fully Observable System

For each setting, the model was trained using a soft target update set at $1e^{-3}$ and a learning rate of $1e^{-5}$. Each model was trained for a total of 100,000 time-steps, with 20,000 time-steps being used for warm-up. Given these training parameters, the training results for the fully observable "smart cities" setting and the fog-node setting are shown in Figure 3. It can be seen that despite the lack of information sharing between the upper and lower intersections, the fog-node setting still performs comparably to a fully observable setting.

However, despite similar training performances, the use of fog nodes results in higher average intersection delay, as shown in Figure 14. Over a 1,000 time-step policy replay, the fully observable model ends with about 150-second average intersection delay. On the other hand, the use of two fog nodes with no communication between them results in almost 300-seconds of average intersection delay at the end of the policy replay.

The primary shortcoming of a fully observable model for traffic signal control problems is that they cannot scale well due to the curse of dimensionality as the number of connected nodes increases. These preliminary results indicate that the use of two separately controlled fog nodes allows for comparable training performance while being more scalable, but at the cost of some performance.

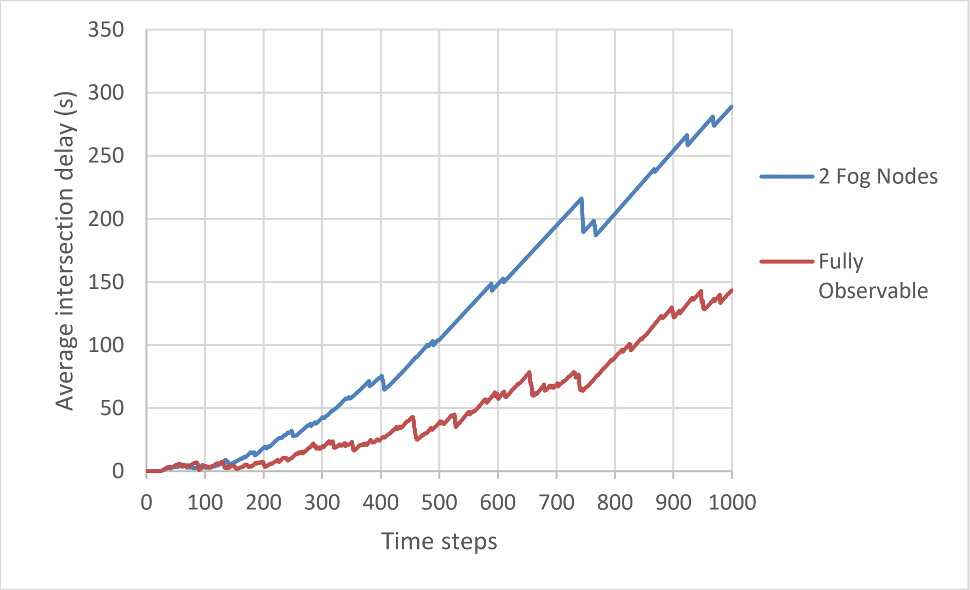


Figure 14: Average intersection delay

CHAPTER 12 CONCLUSIONS

In order to create a more easily scalable, intelligent traffic signal control (TSC) model that can be applied to large networks, this paper proposed the use of graph attention networks (GATs) and fog-node architecture. The added benefit of segmenting large networks into smaller fog-nodes includes the possibility of reducing the number of smart infrastructure units required to facilitate the intelligent TSC models. Multiagent reinforcement learning based models for TSC typically can be affected by the curse of dimensionality. The proposed model addresses scalability in two ways: (i) graph attention that only utilizes relevant node features and neighbor node features to reduce the input complexity, and (ii) fog-nodes that break up the large network into manageable sizes. Preliminary findings show that the proposed model shows promising results that can be scaled into larger networks.

However, their performance in reducing average intersection delay may be relatively inferior compared to a fully observable model. As such, ongoing work on various fog node deployment arrangements and their performance, are expected to provide additional insights on the tradeoff between scalability and performance using the proposed GAT and fog-node architecture. Another promising research direction is to create a simplified or averaged performance within each fog-node to reduce the data size and complexity, thereby allowing fog-nodes to exchange data between each other to make decisions based on other fogs' performances.

CHAPTER 13 SYNOPSIS OF PERFORMANCE INDICATORS

13.1 Part I

Two (2) transportation-related courses were offered annually during the study period that was taught by the PI and a teaching assistant who are associated with the research project. One of these was a newly developed course inspired and directly associated with CCAT research. Three graduate students and a post-doctoral researcher (subsequently designated a Visiting Assistant Professor) participated in the research project during the study period. One (1) transportation-related advanced degree (doctoral) program utilized the CCAT grant funds from this research project, during the study period to support graduate students.

13.2 Part II

Research Performance Indicators: 2 journal articles and 2 conference articles were produced from this project. The research from this advanced research project was disseminated to over 150 people from industry, government, and academia, through 3 conference presentations. These include the 2021 IEEE International Smart Cities Conference (ISC2), the 2021 INFORMS Annual Meeting, and the 2022 ASCE International Conference on Transportation and Development.

One (1) other related research project was funded by a source other than UTC and matching fund sources. At the time of writing, the researchers are still working on developing a specific product (new technologies), procedures/policies, and standards/design practices based on the results of this research project.

Leadership Development Performance Indicators: This research project generated 3 academic engagements and 2 industry engagements. The PI held positions in 2 national organizations that address issues related to this research project. One of the CCAT students who worked on this project holds a leadership position.

Education and Workforce Development Performance Indicators: The methods, data and/or results from this study are being incorporated in the syllabus for the next versions (Fall 2022 and/or Spring 2023) of the following courses at Purdue University: (a) CE 561: Transportation Systems Evaluation, a mandatory graduate level course at Purdue's transportation engineering M.S. and Ph.D. programs, (b) CE 299: Smart Mobility, an optional undergraduate level course at Purdue' civil engineering B.S. program, and (c) CE 398: Introduction to Civil Engineering Systems, a mandatory undergraduate level course at Purdue University's civil engineering program. These students will soon be entering the workforce. Thereby, the research helped enlarge the pool of people trained to develop knowledge and utilize the at least a part of the technologies developed in this research, and to put them to use when they enter the workforce.

Collaboration Performance Indicators: There was collaboration with other agencies, and 1 agency provided matching funds.

The outputs, outcomes, and impacts are described in Chapter 14 below.

CHAPTER 14. STUDY OUTCOMES AND OUTPUTS

14.1 Outputs

14.1.1 Publications, conference papers, or presentations

(a) Journal Papers

Ha, P. Y. J., Chen, S., Du, R., Dong, J., Li, Y., & Labi, S. (2020). Vehicle connectivity and automation: a sibling relationship. *Frontiers in Built Environment, 6, 199.* https://doi.org/10.3389/fbuil.2020.590036

(b) Conference papers

Du, R., Chen, S., Dong, J., Ha, P. Y. J., & Labi, S. (2021, September). GAQ-EBkSP: A DRL-based Urban Traffic Dynamic Rerouting Framework using Fog-Cloud Architecture. In *Proceedings of the 2021 IEEE International Smart Cities Conference (ISC2)* (pp. 1-7). IEEE. https://ieeexplore.ieee.org/abstract/document/9562832

(c) Presentations

Du, R., Chen, S., Dong, J., Ha, P. Y. J., & Labi, S. (2021). GAQ-EBkSP: A DRL-based Urban Traffic Dynamic Rerouting Framework using Fog-Cloud Architecture. In *2021 INFORMS Annual Meeting*, Anaheim, CA, October 24-27, 2021.

Du, R., Chen, S., Dong, J., Ha, P. Y. J., & Labi, S. (2021). GAQ-EBkSP: A DRL-based Urban Traffic Dynamic Rerouting Framework using Fog-Cloud Architecture. In *2021 IEEE International Smart Cities Conference (ISC2)* IEEE Virtual Conference.

Du, R., Chen, S., Dong, J., Ha, P. Y. J., & Labi, S. (2022). A DRL-based urban traffic dynamic rerouting framework with fog-cloud architecture, 2022 ASCE International Conference on Transportation and Development, Seattle, Washington, May 31–June 3, 2022.

14.1.2. Other outputs

The first part of this study addresses the use of fog-cloud architecture for a deep reinforcement learning-based control framework and presents a case study involving urban traffic dynamic rerouting, in a bid to help mitigate traffic congestion at urban areas. The second part of the study recognizes that optimizing traffic signal control at intersections continues to pose a challenging problem particularly for large-scale traffic networks, and uses fog-cloud based multiagent reinforcement learning scalable for controlling urban traffic signal systems. Specifically, the new methodologies, technologies and techniques developed in the study are:

- *A centralized-control system with decentralized execution is built on top of cloud-fog information exchange architecture (cloud-fog-edge):*
  In this project, the cloud serves as a central platform for planning and making decisions at

the system level based on the information collected decentralized by the fog nodes. Rerouting vehicles executing those decisions in a decentralized manner. This arrangement is intended to preserve both the computation capability and the efficiency of information exchange.

- *Attention mechanism combines with deep reinforcement learning:*
  using DRL (deep reinforcement learning) solely in solving problems in the urban traffic systems (with highly dynamic, complex nature and massive information) can be really challenging. However, in this project, we introduced attention mechanism. The attention mechanism helps differentiate the relative importance of input information which enlarge the learning efficiency of the DRL.
- *Combine (OSM) Open Street Map and SUMO (Simulation of Urban Mobility) to build the experiment network:*
  The proposed framework is implemented in a simulation environment using SUMO, which is an open-source simulator with well-defined vehicle parameters. The experiment network is the Manhattan network, which is imported from OSM and cleaned in SUMO.

Other products of this research are as follows:

- A set of analytical models that describe: a centralized-control system with decentralized execution is built on top of cloud-fog information exchange architecture (cloud-fog-edge); uses attention mechanism combined with deep reinforcement learning; and combine (OSM) Open Street Map and SUMO (Simulation of Urban Mobility) to build a network for experimentation.
- Material for the Purdue Graduate course “CE 597 – Artificial intelligence and machine learning for autonomous vehicle operations.”
- Research material and datasets to support future research related to the subjects of multi-level control for CAV and smart infrastructure and AI-based fog-cloud collaboration

14.2 Outcomes

The outcomes of this project are the prospective changes that can be made to the transportation system, or its regulatory, legislative, or policy framework, resulting from research and development outputs. These are:

- Increased understanding and awareness of traffic congestion in large urban areas
- Adoption of new methodology combining DRL with attention mechanism GAT
- Enhanced travel efficiency of large urban networks
- Demonstration of the fog-cloud collaboration concept
- Demonstrate information collection and use motivated by decision contexts

14.3 List of impacts

The impacts of this project are the effects of outcomes on the transportation system, or society in general, such as reduced fatalities, decreased capital or operating costs, community impacts, or environmental benefits. This includes how the research outcomes can potentially improves the operation and safety of the transportation system, increase the body of knowledge and technologies, enlarges the pool of people trained to develop knowledge and utilize new technologies and put them to use, and improve the physical, institutional, and information resources that enable people to have access to training and new technologies. A list of specific impacts from this research project, are as follows:

- Vehicle rerouting is the key to provide better traffic mobility, especially in large urban area. However, only consider local information will not provide optimal routing solution in most cases. In this project, we bring attention mechanism into the framework to help select information based on importance. Thus, the framework can help predict the potential congestion area and help rerouting vehicles avoid future congestion, and thus, enhance the urban mobility. In large urban area, information exchange happens at anywhere and anytime, the efficiency of the information exchange affects the users' decisions and experience. Traditional cloud locates far from users, which brings high latency. By introduce fog nodes into the information exchange architecture, users can get faster respond with low latency. In this project, we built our framework on top of the fog-cloud architecture, which enhance the communication efficiency and gives users (drivers) better experience.
- It is anticipated that the proposed research will provide strong justification for CAV manufacturers, technology companies, and the road agencies to invest in connectivity equipment and facilities, and therefore, will have a higher stake in CAV deployment. Similarly, the need for additional investment in the development and deployment of intelligent infrastructure can be justified. We expect that the research will provide proof that connectivity-equipped AVs and connectivity investments for HDVs can greatly benefit the entire traffic stream in the sense that it will enhance operational efficiency and mobility.
- Justification for wide adoption of 5G/LTE for reduced latency that results in enhanced mobility and safety, especially in the context of large-scale networks such as urban areas
- The impacts of the research will hopefully give a strong justification to both CAV company and DOT's investment in installing connectivity facilities, and that investments in connectivity facilities can greatly benefit the entire transportation system by enhancing mobility and safety.
- We expect that the development of an innovative AI for CAV controls, at large-scale networks comprised of signalized intersections, will yield positive effects on the transport system and society in general. These includes reduced crashes, travel efficiency (reduced travel time) which translate into lower vehicle operating costs, higher economic productivity, and more free time for social activities.
- Six graduate students that worked on this project will enter the workforce in 2023 to help support the workforce that will implement new technologies such as those developed in this study.
- Parts of the research outcomes were incorporated in a graduate level class (the Purdue University course in Spring 2020 and Fall 2021 "CE 597 – Artificial intelligence and machine learning for autonomous vehicle operations, Part I and II." Therefore, the students, who will soon be entering the workforce, benefitted from the outcomes of this research through an academic platform. This helps enlarge the pool of people trained to

develop knowledge and utilize the technologies developed in this research, and to put them to use when they enter the workforce.

REFERENCES

Aazam, M., Zeadally, S., & Harras, K. A. (2018). Fog Computing Architecture, Evaluation, and Future Research Directions. IEEE Communications Magazine, 56(5). doi: 10.1109/MCOM.2018.1700707

Abuelenin, S. M., & Elaraby, S. (2021). A Generalized Framework for Connectivity Analysis in Vehicle-to-Vehicle Communications. IEEE Transactions on Intelligent Transportation Systems. doi: 10.1109/TITS.2021.3052846

al Islam, S. M. A. B., Tajalli, M., Mohebifard, R., & Hajbabaie, A. (2021). Effects of connectivity and traffic observability on an adaptive traffic signal control system. In Transportation Research Record (Vol. 2675, Issue 10). doi: 10.1177/03611981211013036

Arokhlo, M. Z., Selamat, A., Hashim, S. Z. M., & Selamat, M. H. (2011). Route guidance system using multi-agent reinforcement learning. 2011 7th International Conference on Information Technology in Asia: Emerging Convergences and Singularity of Forms - Proceedings of CITA'11. doi: 10.1109/CITA.2011.5999388

Brennand, C. A. R. L., Boukerche, A., Meneguette, R., & Villas, L. A. (2017). A novel urban traffic management mechanism based on FOG. Proceedings - IEEE Symposium on Computers and Communications. doi: 10.1109/ISCC.2017.8024559

Brennand, C. A. R. L., de Souza, A. M., Maia, G., Boukerche, A., Ramos, H., Loureiro, A. A. F., & Villas, L. A. (2016). An intelligent transportation system for detection and control of congested roads in urban centers. Proceedings - IEEE Symposium on Computers and Communications, 2016-February. doi: 10.1109/ISCC.2015.7405590

Cao, A., Fu, B., & He, Z. (2019). ETCS: An efficient traffic congestion scheduling scheme combined with edge computing. Proceedings - 21st IEEE International Conference on High Performance Computing and Communications, 17th IEEE International Conference on Smart City and 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019. doi: 10.1109/HPCC/SmartCity/DSS.2019.00378

Chen, S., Dong, J., Ha, P., Li, Y., & Labi, S. (2021). Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. Computer-Aided Civil and Infrastructure Engineering, 36(7). doi: 10.1111/mice.12702

Ceylan, H. and Bell, M. G. H. (2004). Traffic signal timing optimization based on genetic algorithm approach, including drivers' routing, Transp. Res. Part B Methodol., vol. 38, no. 4, pp. 329–342.

Chu, T., Wang, J. Codeca, L., and Li, Z., 2020Multi-agent deep reinforcement learning for large-scale traffic signal control, IEEE Trans. Intell. Transp. Syst.

Devailly, F.-X. Larocque, D. and Charlin, L. (2021). Ig-rl: Inductive graph reinforcement learning for massive-scale traffic signal control, IEEE Trans. Intell. Transp. Syst., 10.1109/TITS.2021.3070835

Dong, J., Chen, S., Li, Y., Du, R., Steinfeld, A., & Labi, S. (2021). Space-weighted information fusion using deep reinforcement learning: The context of tactical control of lane-changing autonomous vehicles and connectivity range assessment. Transportation Research Part C: Emerging Technologies, 128. doi: 10.1016/j.trc.2021.103192

Dong, J., Chen, S., Li, Y., Ha, P. Y. J., Du, R., Steinfeld, A., & Labi, S. (2020). Spatio-weighted information fusion and DRL-based control for connected autonomous vehicles. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020. doi: 10.1109/ITSC45102.2020.9294550

Dongbin, Z., Dai, Y., and Zhang, Z. (2011). Computational intelligence in urban traffic signal control: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(4), 485-494.

Eliasson, J., & Mattsson, L. G. (2006). Equity effects of congestion pricing. Quantitative methodology and a case study for Stockholm. Transportation Research Part A: Policy and Practice, 40(7). doi: 10.1016/j.tra.2005.11.002

FHWA, Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation, 2020. [Online]. Available: https://ops.fhwa.dot.gov/congestion_report/executive_summary.htm.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). Deep learning, vol. 1, no. 2. MIT Press, Cambridge, MA.

Guanetti, J., Kim, Y., & Borrelli, F. (2018). Control of connected and automated vehicles: State of the art and future challenges. In Annual Reviews in Control (Vol. 45). doi: 10.1016/j.arcontrol.2018.04.011

Guo, Q., Li, L., and Xuegang J.B. Urban traffic signal control with connected and automated vehicles: A survey. Transportation Research Part C: Emerging Technologies 101 (2019): 313-334.

Ha, P. Y. J., Chen, S., Dong, J., Du, R., Li, Y., and Labi, S. (2020a). Leveraging the capabilities of connected and autonomous vehicles and multi-agent reinforcement learning to mitigate highway bottleneck congestion, arXiv Prepr. arXiv2010.05436,.

Ha, P.Y.J., Chen, S., Du, R., Dong, J., Li, Y., & Labi, S. (2020b). Vehicle Connectivity and Automation: A Sibling Relationship. In Frontiers in Built Environment (Vol. 6). doi: 10.3389/fbuil.2020.590036

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-Arnold, G., Osband, I., Agapiou, J., Leibo, J.Z., Gruslys, A. (2018). Deep q-learning from demonstrations, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1.

Ho, M. C., Lim, J. M. Y., Soon, K. L., & Chong, C. Y. (2019). An improved pheromone-based vehicle rerouting system to reduce traffic congestion. Applied Soft Computing Journal, 84. doi: 10.1016/j.asoc.2019.105702

INRIX. (2020). INRIX Global Traffic Scorecard: Congestion cost UK economy £6.9 billion in 2019 . INRIX.

Jiang, J., Dun, C., Huang, T., & Lu, Z. (2018). Graph convolutional reinforcement learning. In arXiv.

Karimi, H., Ghadirifaraz, B., Shetab Boushehri, S. N., Hosseininasab, S. M., & Rafiei, N. (2021). Reducing traffic congestion and increasing sustainability in special urban areas through one-way traffic reconfiguration. Transportation. doi: 10.1007/s11116-020-10162-4

Kipf T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks, in Conference Track Proceedings, 5th International Conference on Learning Representations, ICLR 2017.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Koonce, P. and Rodegerdts, L. (2008). Traffic signal timing manual., United States. Federal Highway Administration.

Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent Development and Applications of {SUMO - Simulation of Urban MObility}. International Journal On Advances in Systems and Measurements, 5(3).

Li, J. Q., Mirchandani, P. B., & Borenstein, D. (2009). Real-time vehicle rerouting problems with time windows. European Journal of Operational Research, 194(3). doi: 10.1016/j.ejor.2007.12.037

Lillicrap, T. P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D. (2016). Continuous control with deep reinforcement learning, in Conference Track Proceedings, 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, May 2-4, 2016.

Lin, L.-J. (1992). Reinforcement learning for robots using neural networks. Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA.

Liu D. and Yang, C. 2019.A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users, IEEE Access, vol. 7, pp. 83120–83136,

Maciejewski, M., & Bischoff, J. (2018). Congestion effects of autonomous taxi fleets. Transport, 33(4). doi: 10.3846/16484142.2017.1347827

Marín-Tordera, E., Masip-Bruin, X., García-Almiñana, J., Jukan, A., Ren, G. J., & Zhu, J. (2017). Do we all really know what a fog node is? Current trends towards an open definition. Computer Communications, 109. doi: 10.1016/j.comcom.2017.05.013

Mnih V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M. (2013). Playing atari with deep reinforcement learning, arXiv Prepr. arXiv1312.5602, 2013.

Nguyen, B. L., Ngo, D. T., Dao, M. N., Bao, V. N. Q., & Vu, H. L. (2021). Scheduling and Power Control for Connectivity Enhancement in Multi-Hop I2V/V2V Networks. IEEE Transactions on Intelligent Transportation Systems. doi: 10.1109/TITS.2021.3091130

Nguyen, B. L., Ngo, D. T., Tran, N. H., Dao, M. N., & Vu, H. L. (2020). Dynamic V2I/V2V Cooperative Scheme for Connectivity and Throughput Enhancement. IEEE Transactions on Intelligent Transportation Systems. doi: 10.1109/tits.2020.3023708
OpenStreetMap contributors. (2017). Planet dump retrieved from https://planet.osm.org . \url{ Https://Www.Openstreetmap.Org }.

Pan, J., Popa, I. S., Zeitouni, K., & Borcea, C. (2013). Proactive vehicular traffic rerouting for lower travel time. IEEE Transactions on Vehicular Technology, 62(8). doi: 10.1109/TVT.2013.2260422

Prashanth L. A. and Bhatnagar, S. (2010). Reinforcement learning with function approximation for traffic signal control, IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, pp. 412–421.

Pupiales, C., Laselva, D., & Demirkol, I. (2021). Capacity and Congestion Aware Flow Control Mechanism for Efficient Traffic Aggregation in Multi-Radio Dual Connectivity. IEEE Access, 9. doi: 10.1109/ACCESS.2021.3105177

Rezaei, M., Noori, H., Rahbari, D., & Nickray, M. (2018). ReFOCUS: A hybrid fog-cloud based intelligent traffic re-routing system. 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation, KBEI 2017, 2018-January. doi: 10.1109/KBEI.2017.8324943

Shapley, L.S. (1953). Stochastic games. Proceedings of the National Academy of Sciences 39, no. 10: 1095-1100.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks, in European semantic web conference, pp. 593–607.

Tang, C., Hu, W., Hu, S., & Stettler, M. E. J. (2020). Urban Traffic Route Guidance Method With High Adaptive Learning Ability Under Diverse Traffic Scenarios. IEEE Transactions on Intelligent Transportation Systems. doi: 10.1109/TITS.2020.2978227

Tesauro, G. (1995). Temporal difference learning and TD-Gammon, Commun. ACM, vol. 38, no. 3, pp. 58–68.

U.S Department of transportation. (2012). Traffic Congestion and Reliability : Trends and Advanced Strategies for Congestion Mitigation. In Office of operation. (Issue September).

U.S. Department of Transportation, B. of T. S. (2020). Transportation Statistics Annual Report 2020. Washington, DC.

van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-Learning. 30th AAAI Conference on Artificial Intelligence, AAAI 2016.

Velicković, P. Cucurull, G. Casanova, A. Romero, A. Liò, P. and Bengio, Y. (2018). Graph attention networks, Conference Track Proceedings, 6th International Conference on Learning Representations, ICLR 2018.

Vinitsky, E.. Parvate, K.. Kreidieh, A.. Wu, C.. and Bayen, A. (2018). Lagrangian Control through Deep-RL: Applications to Bottleneck Decongestion, in IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2018, pp. 759–765.

Wang, S., Djahel, S., Zhang, Z., & McManis, J. (2016). Next Road Rerouting: A Multiagent System for Mitigating Unexpected Urban Traffic Congestion. IEEE Transactions on Intelligent Transportation Systems, 17(10). doi: 10.1109/TITS.2016.2531425

Wang, X. Ke, L. Qiao, Z. and Chai, X. (2021). Large-Scale Traffic Signal Control Using a Novel Multiagent Reinforcement Learning, IEEE Trans. Cybern., 51(1), 174-187.

Wang, Y. Xu, T. Niu, X. Tan, C. Chen, E. and Xiong, H. (2020). STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control, IEEE Transactions on Mobile Computing Conference, IEEE Computer Society, Washington, DC.

Watkins, C. J. C. H. (1989). Learning from delayed rewards, Ph.D. Dissertation, Kings College, London, UK.

Watkins, C. and Dayan, P. (1992): Q-learning. Machine learning 8, no. 3-4 279-292.Webster, F. V. (1958). Traffic signal settings, road research technical paper no. 39. Road Research Laboratory, U.K.

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

Wiering, M. A. (2000). Multi-agent reinforcement learning for traffic light control, in Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000), pp. 1151–1158.

Yang, H., Almutairi, F., & Rakha, H. (2021). Eco-driving at signalized intersections: A multiple signal optimization approach. IEEE Transactions on Intelligent Transportation Systems, 22(5). doi: 10.1109/TITS.2020.2978184

Zhao, J., Mao, M., Zhao, X., & Zou, J. (2021). A Hybrid of Deep Reinforcement Learning and Local Search for the Vehicle Routing Problems. IEEE Transactions on Intelligent Transportation Systems, 22(11). doi: 10.1109/TITS.2020.3003163

Zhu, M., Wang, X., and Wang, Y. (2018). Human-like autonomous car-following model with deep reinforcement learning, Transp. Res. Part C Emerg. Technol., vol. 97, pp. 348–368,

APPENDIX

Published Related Work

Paper 1: Du, R., Chen, S., Dong, J., Ha, P. Y. J., & Labi, S. (2021). GAQ-EBkSP: A DRL-based Urban Traffic Dynamic Rerouting Framework using Fog-Cloud Architecture. In *2021 IEEE International Smart Cities Conference (ISC2) Proceedings*, IEEE.

Abstract

Dynamic rerouting framework can improve urban traffic management by mitigating urban traffic congestion. Emerging technologies such as fog-computing offers low-latency capabilities and facilitates the information exchange between the vehicles and infrastructure systems, and this fosters dynamic rerouting efficiency. In this study, a 2 stage-method combining GAQ (Graph Attention Network- Deep Q Learning) and EBkSP (Entropy Based k Shortest Path) is proposed using a fog-cloud architecture to reroute the vehicles in a dynamic urban environment to achieve improved travel efficiency. First, GAQ analyzes the traffic conditions on each road and for each fog area and assigns a road index based on the information attention from both local and neighboring areas. Second, the route for each vehicle is assigned using EBkSP based on the vehicle priority and route popularity. The results demonstrate attainment of higher speed and lower total travel time for each vehicle in the network, thereby indicating the efficacy of the proposed framework in dynamic rerouting.

Paper 2: Ha, P. Y. J., Chen, S., Du, R., Dong, J., Li, Y., & Labi, S. (2020). Vehicle connectivity and automation: a sibling relationship. *Frontiers in Built Environment*, 6, 199. https://doi.org/10.3389/fbuil.2020.590036

Abstract

The evolution of scientific advances has often been characterized by the amalgamation of two or more technologies. With respect to vehicle connectivity and automation, recent literature suggests that these two emerging transportation technologies can and will jointly and profoundly shape the future of transportation. However, it is not certain how the individual and synergistic benefits to be earned from these technologies is related to their prevailing levels of development. As such, it may be considered useful to revisit the primary concepts of automation and connectivity, and to identify any current and expected future synergies between them. Doing this can help generate knowledge that could be used to justify investments related to transportation systems connectivity and automation. In this discussion paper, we attempt to address some of these issues. The paper first reviews the technological concepts of systems automation and systems connectivity, and how they prospectively, from an individual and collective perspective, impact road transportation efficiency and safety. The paper also discusses the separate and common benefits of connectivity and automation, and their possible holistic effects in terms of these benefits where they overlap. The paper suggests that at the current time, the sibling relationship seems to be lopsided: vehicle connectivity has immense potential to enhance vehicle automation. Automation, on the other hand, may not significantly promote vehicle connectivity directly, at least not in the short term but possibly in the long term. The paper argues that future trends regarding market adoption of these two technologies and their relative pace of advancement or regulation, will shape the future synergies between them.