Center for Advanced Multimodal Mobility Solutions and Education

CAMMSE

Project ID: 2022 Project 17

TRANSIT SIGNAL PRIORITY CONTROL WITH CONNECTED VEHICLE TECHNOLOGY: DEEP REINFORCEMENT LEARNING APPROACH

Final Report

by

Wei Fan (ORCID ID: https://orcid.org/0000-0001-9815-710X)

Tianjia Yang (ORCID ID: https://orcid.org/0000-0002-4392-0419)

Wei Fan, Ph.D., P.E.
Professor, Department of Civil and Environmental Engineering
9201 University City Blvd, Charlotte, NC 28223.
Phone: 1-704-687-1222; Email: wfan7@uncc.edu

for

Center for Advanced Multimodal Mobility Solutions and Education
(CAMMSE @ UNC Charlotte)
The University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223

September 2024

ACKNOWLEDGEMENTS

This project was funded by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE @ UNC Charlotte), one of the Tier I University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (USDOT), under the FAST Act. The authors are also very grateful for all of the time and effort spent by DOT and industry professionals to provide project information that was critical for the successful completion of this study.

DISCLAIMER

The contents of this report reflect the views of the authors, who are solely responsible for the facts and the accuracy of the material and information presented herein. This document is disseminated under the sponsorship of the U.S. 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. The contents do not necessarily reflect the official views of the U.S. Government. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Transit Signal Priority (TSP) is a traffic signal control strategy that can provide priority to transit vehicles and thus improve transit service. However, this control strategy generally causes adverse effects on other traffic, which limits its widespread adoption. The development of Connected Vehicle (CV) technology enables the real-time acquisition of fine-grained traffic information, providing more comprehensive data for the optimization of traffic signals. Simultaneously, optimization algorithms in the field of TSP have been advancing at a rapid pace. Artificial intelligent (AI)-powered techniques, such as Deep Reinforcement Learning (DRL), have become promising approaches for addressing TSP problems recently.

In this study, we develop adaptive TSP control frameworks for both isolated intersection scenarios and multiple intersection scenarios, assuming the implementation of CV technology. Leveraging the comprehensive traffic data obtained from CVs, our frameworks employ both single-agent DRL and multi-agent DRL techniques to address optimization problems. The controllers, based on our proposed frameworks, are tested in simulation environments and compared with various widely used traffic signal controllers across different scenarios.

Results show that in the isolated intersection scenarios, the proposed DQN controller has the best performance in terms of average person delay. Compared to the pretimed signal controller, it reduces the average person delay by 18.77% in peak hours and 23.37% in off-peak hours. Furthermore, it also results in decreased average delays for both buses and cars. The sensitivity analysis results indicate that the proposed controller has the potential for practical applications, as it can effectively handle some dynamic changes. Furthermore, the corridor-level experimental results demonstrate that the proposed controller, MAPPO-M, which adopts the multi-agent proximal policy optimization (MAPPO) algorithm and the multi-discrete action space, exhibits superior performance in terms of bus services while maintaining decent service for regular traffic. Additionally, sensitivity analysis indicates that MAPPO-M can achieve its best performance when the CV market penetration rate exceeds 60%. It is also capable of handling dynamics introduced by varying passenger occupancy and bus arrival headways.

Chapter 1. Introduction

1.1. Problem Statement

As urbanization and population growth continue, travel demand continues to rise. However, the growth rate of transportation infrastructure supply, especially in metropolitan areas, is low, leading to a significant increase in traffic congestion. In general, there are two options to address this issue: One is to build more transportation infrastructure, and the other is to improve the efficiency of the existing transportation system. Given the limitations of space and funding, improving efficiency is the more realistic choice. As a result, public transportation, which is more efficient than private transportation, is gaining prominence in the urban transportation system. However, users of public transportation often have to share spaces and experience longer travel times, which makes it less attractive than private transportation. To this end, transit priority strategies, which can greatly help in developing a more sustainable, equitable, and efficient urban transportation system, have been extensively studied. The implementation strategies include the formulation of policies to prioritize public transportation, the provision of financial subsidies for public transportation, the construction of high accessible public transportation system, and the granting of priority to transit vehicles, etc. Among them, transit signal priority (TSP) is a critical operational strategy that can improve the service performance of transit vehicles on the road.

TSP generally adjusts the signal plan to ensure priority for transit vehicles at intersections, arterials, or networks (Skabardonis, 2000). However, this control strategy generally causes adverse effects on other traffic, which limits its widespread adoption. In order to solve this problem, adaptive TSP, which can mitigate negative effects while still providing priority to transit vehicles, has been studied for decades (Christofa & Skabardonis, 2011; Ma et al., 2010; Skabardonis & Geroliminis, 2008). Generally, adaptive TSP has to obtain real-time traffic data to optimize the traffic signal plan. Traditional traffic data sensors, such as loop detectors, cameras, and radars, are installed in fixed positions and are therefore more or less deficient in acquiring real-time data. Recently, with the rapid development of connected vehicle (CV) technology, more accurate and more comprehensive real-time traffic data can be easily obtained. This advantage can surely boost the advancement of adaptive TSP, and many researchers have integrated CV technology with adaptive TSP (Ghanim & Abu-Lebdeh, 2015; Zeng et al., 2021). The U.S. Department of Transportation (USDOT) has also included TSPCV on its list of High-Priority Applications and Development Approaches (U.S. Department of Transportation, 2011).

CV technology refers to the integration of wireless communication technology, such as dedicated short-range communication and cellular technology, in vehicles, enabling them to communicate with other vehicles, infrastructure, and other traffic participants within a certain distance (Guo et al., 2019). With the advent of CV technology, real-time detailed information, such as passenger occupancy, can be obtained. This enables the adoption of more fine-grained metrics, such as average person delay, as the objective for optimizing traffic signals. As a result, transit vehicles with more passengers can cross intersections more efficiently, reducing travel time for passengers and encouraging greater usage of public transportation services. Furthermore, the availability of rich real-time traffic data provided by CV technology opens up the possibility of optimizing traffic signal controllers through data-driven approaches.

Optimization algorithms in the field of traffic signal control (TSC) have been advancing at a rapid pace. Among them, deterministic algorithms such as mixed-integer nonlinear programming (MINLP) and dynamic programming (DP) have been widely used to optimize traffic signal control (Feng et al., 2015; Li & Ban, 2019; Priemer & Friedrich, 2009). However, these algorithms have to model the traffic environment as comprehensively as possible, which is computationally intensive, time-consuming, and thus impractical (Mohamad Alizadeh Shabestary, 2019). On the other hand, conventional stochastic algorithms, like genetic algorithms (Lee et al., 2006; Teklu et al., 2007; Yang & Fan, 2023), tend to get stuck in sub-optimal solutions, making them unreliable for real-world implementation. Due to the availability of real-time traffic data in CV environments, reinforcement learning (RL) algorithms, which are data-driven and can learn the optimal control strategies when interacting with the environment, have gained significant attention as potential solutions to optimize TSC problems (Aslani et al., 2017; Chow et al., 2021; Li et al., 2016). RL was initially developed to solve problems with discrete states and actions. However, when integrated with deep learning, the method is commonly referred to as deep reinforcement learning (DRL) and becomes a promising approach for TSC problems (Genders & Razavi, 2016; Mao et al., 2023; Shabestary & Abdulhai, 2022).

Most of the existing DRL studies have focused on optimizing the signal control problem that only considers the purely private traffic mode. This study, however, seeks to propose a robust adaptive TSP controller in a CV environment that grants priority to transit vehicles while minimizing the negative impact on regular traffic. DRL approaches will be employed in this research to solve the signal control optimization problem. Comprehensive simulation experiments based on real-world traffic configurations will be conducted to evaluate the effectiveness of the proposed control algorithms. This study contributes to the development of adaptive TSP controllers in the CV environment by utilizing advanced learning-based optimization approaches.

1.2. Objectives

The objectives of this study are to:

1) Conduct a comprehensive literature review on TSC and TSP related optimization algorithms.
2) Propose adaptive TSP control systems by applying the DRL approach to solve optimization problems on two different levels: isolated intersections and corridors.

3) Build simulation testbeds based on both hypothetical and real-world traffic configurations.
4) Conduct comprehensive simulation experiments to evaluate the effectiveness of the proposed control systems.
5) Analyze and discuss the simulation results in different scenarios.

1.3. Report Overview

The report is organized as follows. A comprehensive literature review is presented in Chapter 2. The methodology used in DRL-based TSP controllers is described in Chapter 3. Chapters 4 and 5 provide details about the traffic configurations, simulation settings, and analysis results related to isolated intersection scenarios and corridor scenarios, respectively. Finally, in Chapter 6, the conclusions from this study are summarized and the future work is suggested.

Chapter 2. Literature Review

2.1. Introduction

This chapter provides a comprehensive review of the development of TSC and TSP related optimization algorithms. The following sections are organized as follows. Section **Error! Reference source not found.** discusses the existing research on conventional algorithms. The development of reinforcement learning algorithms utilized in TSC and TSP research is reviewed in section **Error! Reference source not found.**. Finally, a summary of the chapter is given in section 2.4.

2.2. Conventional Optimization Algorithms

2.2.1. Deterministic Algorithms

The majority of conventional optimization algorithms are deterministic. In the TSC domain, these algorithms require the TSC system to be modeled as comprehensively as possible, which is often computationally intensive and time-consuming. As a result, the significant challenge when utilizing deterministic algorithms is to balance the complexity of the algorithm with the practical value of the controller, especially in dynamic real-world conditions.

Feng et al. (2015) proposed an adaptive traffic signal controller that can optimize the signal phase and timing in a connected vehicle environment. A two-level optimization problem was formulated, and dynamic programming (DP) was employed to solve this problem. Given the low CV market penetration rate, a vehicle state estimation model based on the traffic data obtained via CVs was developed to provide complete information on vehicles approaching the intersection. The performance of the proposed controller was evaluated by modeling a real-world isolated intersection in VISSIM. Results showed that the proposed controller was more effective than a well-tuned fully actuated controller in reducing total delay by as much as 16.33% at a 100% CV market penetration rate and exhibited similar performance at a 25% market penetration rate.

Li and Ban (2019) proposed a signal timing controller for optimizing the signal timing at an isolated intersection with a fixed cycle length. The controller utilized vehicle arrival data obtained via CV technology as input to find optimal green time durations, with the objective of minimizing the weighted sum of vehicle fuel consumption and travel time. The optimization problem was formulated as a mixed-integer nonlinear program (MINLP), which was then solved by decomposing it into a sequential of signal-stage timing decision problems using dynamic programming. A stage in these sequential problems was referred to as a signal phase. Simulation experiments were conducted in VISSIM to evaluate the performance of the proposed model. The results indicated that the proposed controller outperformed the actuated controller under all scenarios. Additionally, in terms of computational times for solving the MINLP problem, the DP method outperformed the NOMAD solver in MATLAB, especially for large-scale problems.

He et al. (2014) proposed a multimodal traffic signal controller that can handle multiple active priority requests while ensuring signal coordination and vehicle actuation in the corridor, under the condition that V2I technology is available. The optimization problem was formulated as a request-based mixed-integer linear program (MILP) that simultaneously considered multiple priority requests, coordination, and real-time actuation. Numerical experiments were conducted in VISSIM based on a real-world two-intersection corridor to test the effectiveness of the proposed controller. The results showed that, compared to the coordinated-actuated traffic signal controller with TSP, the proposed controller reduced bus delay and pedestrian delay by 24.9% and 14%, respectively, in high traffic demand scenarios, while providing similar performance in terms of passenger car delay. In the meantime, real-time actuated control was maintained.

2.2.2. Stochastic Algorithms

Given that TSC systems are large, complex, nonlinear, and stochastic in nature (Dongbin et al., 2012), stochastic algorithms, which are mostly model-free, are more likely to provide feasible solutions to TSC problems. However, conventional stochastic algorithms, such as metaheuristic algorithms, have their limitations. They tend to converge to suboptimal solutions, and the decision-making process can also be time-consuming. These disadvantages make these algorithms less reliable for real-world applications.

Lee et al. (2006) developed a real-time adaptive traffic signal control system composed of a genetic algorithm (GA) optimization module, an internal traffic simulation module, and a database management module. This system operated in an acyclic rolling horizon real-time manner to control traffic signals in an arterial with three intersections. Simulation experiments were conducted in PARAMICS, considering three scenarios with different levels of traffic demand, namely high, medium, and low. The performance of the proposed signal control system was analyzed, and the results showed that it performed efficiently in all scenarios. For example, when compared to the pre-timed controller, the proposed system reduced the total vehicle delay by 12.9% in the high-demand scenario. Moreover, the system significantly reduced the delay standard deviations in high and medium-demand scenarios.

Ghanim and Abu-Lebdeh (2015) presented an innovative real-time traffic signal control system utilizing a combination of a GA optimizer and an artificial neural network (ANN). The optimizer was responsible for optimizing traffic signal timing with TSP control, while the ANN predicted bus arrival times taking into account dwell times at bus stops. The authors evaluated six different signal control systems using VISSIM on a two intersecting one-way network with four bus stops. Results showed that the proposed signal control system significantly reduced traffic delay and stops by up to 90%. Regarding transit traffic, it had the capability to reduce transit delay and number of stops, varied by 15% to 85%, depending on the traffic demand and control type. Importantly, experimental results indicated that the proposed signal control system did not have an adverse impact on crossing traffic.

García-Nieto et al. (2012) developed a network-wide signal control system that can optimize the duration of each phase in all traffic lights in an entire urban road network. The system utilized a particle swarm optimization (PSO) algorithm to maximize the number of vehicles reaching their destinations and minimize the total travel time. Two road networks located in metropolitan areas of different cities were modeled in SUMO to evaluate the performance of the proposed system. Results indicated that the proposed system outperformed two other signal control systems, namely the SUMO cycle programs generator and a random search algorithm. The system demonstrated improvements in the number of vehicles that reach their destinations as well as the mean travel time.

2.3. Reinforcement Learning Algorithms

The Markov Decision Process (MDP) is a mathematical framework usually used to model sequential decision-making problems where an agent interacts with an environment to maximize the reward (Sutton & Barto, 2018). The TSC problem can be formulated as an MDP and RL algorithms are well-suited for solving MDPs because they learn through trial and error by continuously updating the policies based on the rewards received. This allows the agent to adapt to changing states and make better decisions over time. Besides, due to the availability of real-time traffic data in CV environments, RL algorithms, as data-driven approaches, have gained significant attention as potential solutions to optimize TSC problems. The operation of an RL algorithm typically involves the following steps: observation of the current state of the environment, selection of an action based on the current policy, receipt of a reward from the environment, and transition to the next state. According to the received reward, the agent iteratively updates its policy to eventually achieve an optimum control policy.

2.3.1. Reinforcement Learning

RL has been utilized in TSC research since the mid-1990s, with a significant increase in the publication of research papers starting in 2010. The performance improvements provided by the use of RL as an optimization approach are compelling even in the initial stage of implementation (Mannion et al., 2016).

Thorpe and Anderson (1996) proposed a traffic signal controller with the goal of minimizing the time taken for a fixed number of vehicles to traverse a 4 x 4 grid road network. To achieve this, they utilized SARSA, an RL algorithm, that employed replace traces and greedy action selection in the controller. The RL agent was modeled using three different state representations, namely, the vehicle counts representation, the fixed distance representation, and the variable distance representation. Simulation experiments were conducted to test the performance of the proposed controller. Results indicated that it could learn signal control strategies that approached the optimal performance. The most effective state representation was found to be the fixed or variable distance methods.

Abdulhai et al. (2003) presented a case study using Q-learning, one of the most popular RL algorithms, to control the traffic signal at an isolated intersection. The state information provided to the proposed RL agent included queue lengths on the four approaches and the elapsed phase time. The action was defined as the choice to remain or switch the current phase. The reward for the RL agent was the total vehicle delay in the queue incurred between the successive decision points. The findings of the study revealed that the proposed RL agent exhibited superiority compared to the pre-timed signal controller, especially in scenarios where traffic demand varied over time. This was attributed to the ability of the RL agent to adapt to fluctuations in traffic flow.

El-Tantawy and Abdulhai (2010) proposed an acyclic adaptive signal control system that utilized Q-learning to optimize the signal plan. The action defined in the RL agent was the selection of the phase index, allowing for variable phase sequences in the signal controller. The reward was defined as the change in the total summation of the cumulative delay for all vehicles in the system. Furthermore, three different state representations were defined, namely, the arrival of vehicles in the current green direction and queue length in the red direction, the queue length, and the cumulative delay. To evaluate the performance of the proposed controller, a real-world intersection located in downtown Toronto and the traffic volume obtained in the morning peak hour were modeled in a simulation environment. The performance of the proposed Q-learning signal controller was compared to a pre-timed controller optimized using the Webster method. The results showed that the proposed controller consistently outperformed the pre-timed signal controller, regardless of the state representations and traffic demand conditions. Additionally, the cumulative delay representation proved to be superior to other state representations in high-demand scenarios.

2.3.2. Deep Reinforcement Learning

RL was originally proposed to solve problems with discrete states and actions. However, it may become less effective in addressing TSC problems that have large state and action spaces. The integration of RL with deep learning, referred to as DRL, offers a promising approach to TSC optimization. In recent years, numerous research papers have been published in this area, highlighting the potential of DRL in TSC optimization.

Wei et al. (2018) introduced an intelligent traffic signal controller that utilized Deep Q-Network (DQN) with modifications named Phase Gate and Memory Palace. The state in this study was a combination of various factors, including the queue length, number of vehicles, updated waiting time of vehicles, an image representation of vehicles’ positions, current phase, and next phase. The action was defined as the selection of whether to keep or change the current phase. The reward was defined as a weighted sum of total queue length, total delay, total waiting time, an indicator of phase switches, total number, and total travel time of vehicles that passed the intersection. Evaluation experiments were conducted on the simulation platform SUMO using both synthetic and real-world traffic demand data. Results showed that the proposed controller outperformed the other three controllers named pre-timed controller, self-organizing traffic light controller, and DRL for traffic light controller. Additionally, the authors investigated the policies learned from the real-world data and demonstrated that the proposed DRL algorithm could effectively accommodate the changes in traffic demand in the real world.

Liang et al. (2019) proposed a DRL traffic signal controller that utilized a modified DQN algorithm to control the SPaT. To enhance its performance, the authors incorporated various techniques, including dueling network, target network, double Q-learning network, and prioritized experience replay, into the DQN agent. The state was defined as an image-like input that consisted of two matrices representing the position and speed of vehicles approaching the intersection. The action was defined as how to change the duration of every phase in the next cycle. The reward was defined as the change in the cumulative waiting time between consecutive cycles. The performance of the proposed controller was evaluated using SUMO, and results showed that it could reduce the average waiting time by more than 25% compared to the pre-timed controller. Moreover, the modified DQN agent outperformed the conventional DQN agent in terms of learning speed and other metrics.

Shabestary and Abdulhai (2022) proposed an innovative adaptive traffic signal controller that utilizes real-time traffic data obtained through CV technologies. This controller is capable of handling unprocessed, high-dimensional traffic data from CVs and is self-learning. A DQN agent with a convolutional neural network was developed to minimize vehicle delays. The real-time position and speed of CVs were preprocessed into an image-like structure that consisted of two same-sized matrices, along with the elapsed time, which was then used as the state in the DQN agent. The action space included all possible phases, each of which was a combination of nonconflicting movements. The reward was defined as the reduction of cumulative delay in consecutive time steps. The authors conducted comprehensive experiments to evaluate the performance of the proposed controller, and the results showed that it outperformed other alternatives, including pre-timed, actuated, and conventional Q-learning controllers. Furthermore, the results demonstrated the generalization and robustness of the proposed controller to some extent.

2.3.3. Multi-Agent Reinforcement Learning

In fact, intersections are not isolated from each other, the control for one intersection will impact other intersections in the network. Since RL-based signal controllers exhibit superior performance at isolated intersection scenarios, one approach is to train a centralized agent to control the whole network. However, it is hard for a centralized agent to scaler to a large network. To address the scalable issue, a feasible way is to implement multiagent reinforcement learning (MARL) algorithms. The road network contains multiple intersections, it can be formulated as a multi-agent system, each agent controls a single intersection or a subgroup of intersections. Recent years, with the rapid advancement of MARL algorithms, many researchers have been working on applying these sophisticated algorithms to multiple intersection scenarios.

Song and Fan (2023) introduced an innovative traffic signal control framework that integrates MARL algorithms for traffic control with CAV platooning techniques for vehicle control. The integration is designed to improve the overall traffic performance along corridors. The MARL algorithm utilized in this study was the state-shared MADQN. Assuming the presence of infrastructure to infrastructure (I2I) communication technology, each intersection can share its state with adjacent intersections. Therefore, the input state for each agent is a matrix including the state of the ego intersection along with the states of its neighboring intersections. This information sharing mechanism can ensure a certain degree of signal coordination between the intersections. Additionally, the CACC technique is leveraged to facilitate the formation of platoons among CAVs, thereby further enhancing traffic efficiency. A testbed corridor with seven intersections is built based on real-world traffic configurations. The results demonstrated the superiority of the proposed framework over alternative approaches, such as fixed-time control and actuated control. Notably, the integration of shared-state MARL and CAV platooning further enhances the performance compared to deploying these technologies separately.

Mao et al. (2023) proposed a multi-agent attention-base soft actor-critic (MASAC) model to control the traffic signals along arterials. They use MASAC method to search for more solution space. Besides, the attention mechanism is also being integrated into their model to extract enriched traffic information. To assess the efficacy of their proposed mode, three hypothetical arterials were built using SUMO. Results showed that their proposed model outperformed other approaches, including the multiband-based method and various DRL algorithms. They also conducted comprehensive ablation experiments to investigate the contribution of each component within their model. The findings demonstrated the substantial impact of attention mechanisms on performance enhancement. Interestingly, the study revealed that the communication module might not be useful when employing the centralized training technique.

With the goal of improving the services for both cars and buses, Yu et al. (2023) proposed a traffic signal controller enhanced by the MARL framework. The novelty of their contribution lies in the design of a unique reward function capable of simultaneously minimizing total vehicle delays and homogenizing bus headways. There are two essential components in the reward function, one reflecting the car traffic efficiency and the other representing the efficiency of the bus system. An adjustable weight coefficient is introduced to balance the performance of cars and buses. DQN, which is a very popular DRL algorithm in TSC research area, is utilized in this study. The authors firstly explored the tradeoff between car and bus traffic performance by varying the weight coefficient in the reward function, identifying the optimal value for the weight. Subsequently, extensive experiments were conducted to validate the superiority of the proposed controller. More importantly, unlike general research in this field, the authors used different networks for training and testing, demonstrating the scalability and transferability of their proposed controller.

Table 2-1 Literature Review on Optimization Algorithms in TSC

Work	**Algorithm**	**Scenario**	**Simulator**	**Result comparison**
Thorpe and Anderson (1996)	SARSA	4 x 4 grid network	Not specified	Pre-timed controller
Abdulhai et al. (2003)	Q-learning	Isolated intersection	Not specified	Pre-timed controller
Lee et al. (2006)	GA	Three-intersection corridor	PARAMICS	Pre-timed controller;
El-Tantawy and Abdulhai (2010)	Q-learning	Isolated intersection	PARAMICS	Pre-timed controller
García-Nieto et al. (2012)	PSO	A 0.75km^2 network in a metropolitan	SUMO	Pre-timed controller; Random search controller
He et al. (2014)	MILP	Two-intersection corridor	VISSIM	Actuated coordination controller
Ghanim and Abu-Lebdeh (2015)	GA	A two intersecting one-way network	VISSIM	Pre-timed controller; Actuated controller
Feng et al. (2015)	DP, Enumeration	Isolated intersection	VISSIM	Actuated controller

Wei et al. (2018)	Modified DQN	Isolated intersection	SUMO	Pre-timed controller;
				Actuated controller;
				DQN controller

Li and Ban (2019)	MINLP, DP	Isolated intersection	VISSIM	Actuated controller

Liang et al. (2019)	double dueling DQN	Isolated intersection	SUMO	Pre-timed controller;
				Adaptive controller;
				DQN controller

Shabestary and Abdulhai (2022)	DQN	Isolated intersection	PARAMICS	Pre-timed controller;
				Adaptive controller;
				Q-learning controller

Song and Fan (2023)	MADQN	Seven-intersection corridor	SUMO	Pre-timed controller; Actuated controller
Mao et al. (2023)	MASAC	Three-intersection corridor, six-intersection corridor, ten-intersection corridor.	SUMO	multiband-based controllers; DRL-based controllers
Yu et al. (2023)	Independent DQN	Five-intersection corridor, ten-intersection	SUMO	Pre-timed controller;

		corridor, two crossing corridors with nine intersections.		Longest queue first controller; Max pressure controller; Centralized RL-based controller

2.4. Summary

A comprehensive review and synthesis of the current state-of-the-art research related to TSC and TSP have been discussed and presented in the preceding sections. This is intended to provide a solid reference and assistance for selecting optimization methods for transit signal priority stratiges and for developing effective adaptive signal controllers for future tasks.

Chapter 3. Methodology

3.1. Single-Agent Reinforcement Learning

3.1.1. Markov Decision Process Formulation

The Markov Decision Process (MDP) is a mathematical framework usually used to model sequential decision-making problems where an agent interacts with an environment to maximize the reward signal. During the interaction, the agent takes actions based on the current state of the environment, and in response, the environment presents a new state as well as a reward (Sutton & Barto, 2018). The traffic signal control problem can be formulated as an MDP in which the state, action, and reward are properly defined. The interaction between the traffic signal control agent and the traffic environment can be mathematically described by a five-tuple $\langle S, A, P, R, \gamma \rangle$, where S (state space) generally represents the set of traffic information obtained from the environment, A (action space) represents the possible operations to control the SPaT, P is the state transition matrix determining the next state based on the current state and action, R is the reward received from the environment after taking the action, and γ is the discount factor. To formulate the TSC problem as MDPs, it is essential to properly define the state space, action space, and reward function. The state includes information such as queue length, cumulative waiting time, number of vehicles per lane, and phase duration. Actions can correspond to different signal control strategies, such as selecting possible green phases, keeping or changing the current phase, and updating the phase duration with a predefined length. The reward function can be defined to reflect various objectives, e.g., minimizing delays, reducing fuel consumption, and improving safety (Haydari & Yilmaz, 2022). Reinforcement learning algorithms are well-suited for solving MDPs because they learn through trial and error by continuously updating the policies based on the rewards received. This allows the agent to adapt to changing states and make better decisions over time. The operation of an RL algorithm typically involves the following steps: observation of the current state of the environment, selection of an action based on the current policy, receipt of a reward from the environment, and transition to the next state. According to the received reward, the agent iteratively updates its policy to eventually achieve an optimum control policy.

3.1.1.1. State space

Two types of state spaces are used in this study: the vehicle-based array state, and the combined state consisting of a vehicle-based array and a feature-based vector, as illustrated in Figure 3-1.

For the vehicle-based array state space, the input traffic states used in the study are the passenger occupancy and speed of CVs approaching the intersection, which is formatted into image-like representations by using the discrete traffic state encoding (DTSE) method. DTSE is favored as it offers the most complete traffic information at the intersection. Additionally, real-time, high-resolution traffic data can be easily obtained via CV technology.

Specifically, each approaching lane within a certain distance L_{lane} from the intersection stop line is discretized into small cells with a specific length d, which is usually the average headway distance between stopped vehicles. The state array is formed by assigning the information of each vehicle to the corresponding cells, as illustrated in Figure 3-1. The location of each vehicle is identified based on the position of its head. Therefore, even when a vehicle covers multiple cells on the road, there will be no problem of being recognized in multiple cells. The state array consists of two tables, resulting in a state space of $2 \times (L_{lane}/d) \times N_{lane}$. One table is used to store the passenger occupancy and the other is used to store the speed of each CV. This vehicle-based information can be obtained via CV technology. Please note that the vehicle-based array only contains information from CVs, as the controller cannot communicate with non-CVs (NCVs) to gather their information.

The combined state space utilizes fusion data obtained from multiple data sources. A feature-based vector is combined with the previously defined vehicle-based array. The vector has a length of N_{lane} corresponding to the number of lanes. In this study, the feature value used in the vector is the number of queued vehicles in each lane. This information is assumed to be extractable from images captured by cameras located at the intersection, leveraging the advancements of computer vision techniques in the field of transportation.

In the isolated intersection scenario, a typical four-approach intersection with four lanes in each approach is investigated. To capture the traffic information effectively, we set the detection range L_{lane} to 350 meters and the cell length d to 7 meters. Therefore, the vehicle-based array has a size of 2*50*16, and the feature-based vector has a size of 16.

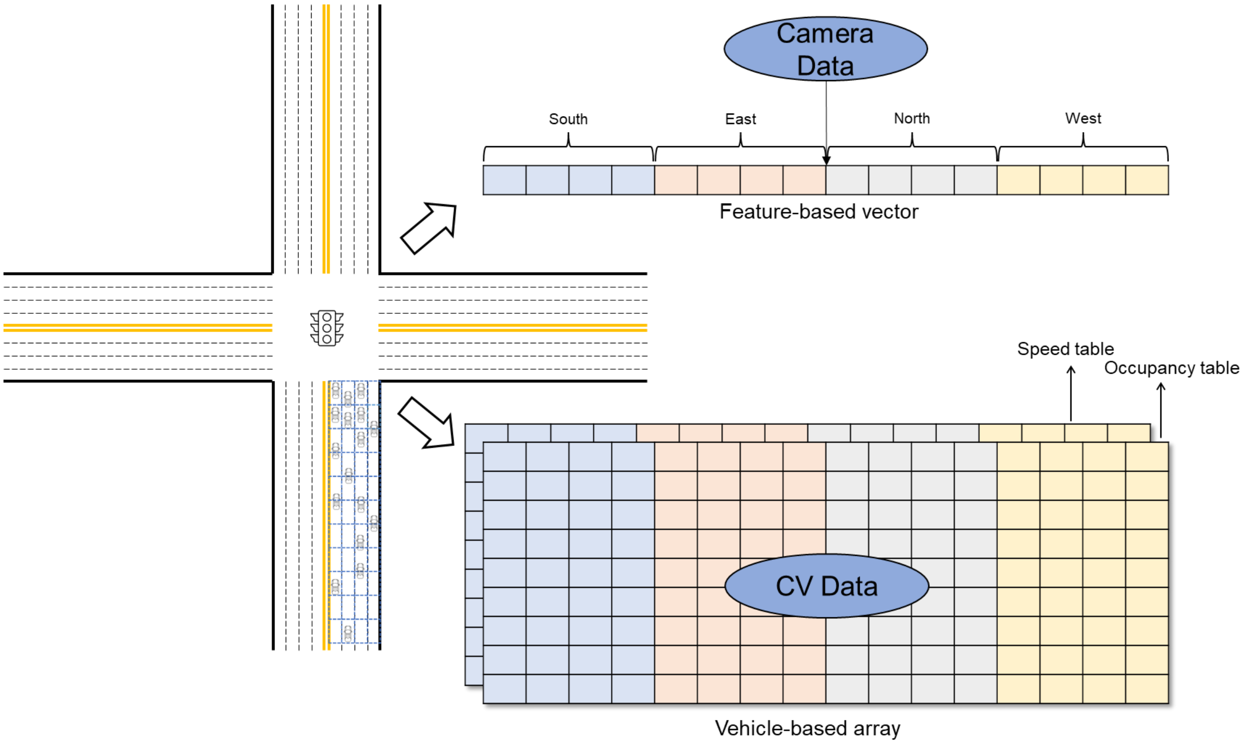


Figure 3-1 Illustration of State Representations for Isolated Intersection in the Study

3.1.1.2. Action space

In this study, two types of action spaces are defined: discrete action space and multi-discrete action space. The discrete action space only includes possible phases, while the multi-discrete action space contains both possible phases and timings.

Except for right-turn movements, there are eight traffic movements in a typical four-approach intersection, namely east through (E), west through (W), east left-turn (EL), west left-turn (WL), north through (N), south through (S), north left-turn (NL), and south left-turn (SL). These movements can be combined into eight non-conflicting movements, corresponding to eight valid phases. Each phase indicates that the corresponding movements will be set to green, while other movements except right-turn movements will be set to red. Right turns are permitted all the time with a lower right-of-way.

The discrete action space has eight actions, representing the eight different phases： $A = \{(NL, SL), (N, NL), (S, SL), (S, N), (EL, WL), (E, EL), (W, WL), (E, W)\}$. At each decision step, an action is selected from this set. If the phase represented by the action is the same as the current phase, the green time is extended by one second. Otherwise, the signal is switched to the chosen phase. Please note that the switching operation includes a transition time, which includes the yellow time, all-red time, and minimum green time. During the transition time, the signal controller remains on hold and does not take any action. Otherwise, it makes decisions every second. In this study, the yellow time is set to 3 seconds, the all-red time is set to 2 seconds, and the minimum green time is set to 5 seconds.

The multi-discrete action space is defined as the cartesian product of two discrete action spaces, denoted as $A = \{(n,t)|n \in N_{phase} \text{ and } t \in T_{duration}\}$. The N_{phase} represents the set of valid phases and is the same as the previously mentioned action space, $N_{phase} = \{0,1,\dots,7\}$. The $T_{duration}$ represents the set of time durations for determining the green time of the selected phase. In this study, the phase durations range from 0 to 45 seconds, which aligns with the range used in the ASC. Considering that the minimum green time is set to 5 seconds, the range of valid phase durations, denoted as $T_{duration}$, is defined as $\{0,1,\dots,40\}$. Therefore, the multi-discrete action space defined in this study is $A = \{(n,t)|n \in \{0,1,\dots,7\} \text{ and } t \in \{0,1,\dots,40\}\}$. At each decision step, a two-element tuple is selected. The first element indicates the phase, and the second element indicates the duration of that phase. The phase-switching logic remains the same as in the discrete action space. The key distinction is that in this action space, the action not only determines the next phase but also specifies the duration of that phase. Therefore, the decision-making frequency is significantly reduced.

3.1.1.3. Reward function

In the field of traffic signal control research, various reward functions have been utilized, including the negative number of vehicles in queues, the negative cumulative queue length, and the negative cumulative delay. In this study, the reward function used is the reduction in cumulative person delay between sequential decision steps.

$$CPD^k = \sum_{n \in S^k} d_n^k * O_n^k \tag{3.1}$$

$$r^k = CPD^{k-1} - CPD^k \tag{3.2}$$

Where k represents the current decision step. n is the CV index. S^k denotes the set of CVs at decision step k. CPD^k means the cumulative person delay in decision step k. r^k denotes the reward in decision step k. At each decision step k, the controller obtains S^k, which is the set of CVs approaching the intersection within the given distance L_{lane}, and d_n^k, which represents the delays of CV n at step k, as long as O_n^k, which represents the passenger occupancy of CV n. The cumulative person delay of CVs at step k, denoted as CPD^k, is calculated using equation 3.7. The cumulative person delay of CVs at step $k-1$, denoted as CPD^{k-1}, is stored in the controller, and the reward at step k, denoted as r^k, is calculated using equation 2. It is worth noting that when the passenger occupancy of all CVs is set to 1 regardless of their vehicle type, the DRL agent is a typical traffic signal controller without TSP.

3.1.2. Deep Q-Network

Deep Q-Network (Mnih et al., 2015), as far as I know, is the most popular DRL algorithm used in the field of traffic signal control. DQN is a value-based RL algorithm, where the state-action value function, known as the Q-function, plays a critical role. The Q-function evaluates the quality of a given action in a particular state. The optimal Q-function can be expressed by the following equation.

$$Q^{*}\,(s,a) = max_{\pi}\mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s, a_t = a, \pi] \qquad (3.3)$$

Where s refers to the current state, a is the current action, r is the reward. The $Q^{*}\,(s,a)$ is the maximum value of the state-action pair (s,a) as determined by the policy π. This value is calculated by summing the present values of all future rewards in each time step t. To determine the present value of a future reward, a discount rate denoted as γ, is introduced.

The optimal Q-function follows an important principle known as the Bellman equation.

$$Q^{*}\,(s,a) = \mathbb{E}_{s'}[r + \gamma max_{a'} Q^{*}\,(s',a') | s,a] \qquad (3.4)$$

This equation is straightforward. If we know $Q^{*}\,(s',a')$, which represents the optimal value of the next state-action pair (s',a'), then $Q^{*}\,(s,a)$ can be achieved by selecting the action that maximizes the expected value of $r + \gamma Q^{*}\,(s',a')$.

By iteratively using the Bellman equation, the optimal Q value can be estimated. However, in many scenarios, it may be impractical to employ this equation for value iteration. For example, when the state/action space is large for some real-world problems, the value iteration process can become computationally intensive. Therefore, algorithms that utilize functions, such as linear and non-linear functions, to approximate the Q-function have been developed. When a deep neural network with weights θ, such as the deep convolutional neural network used in this study, is employed to approximate the Q value, it is referred to as a Deep Q-network.

$$Q(s,a;\boldsymbol{\theta}) \approx Q^{*}\,(s,a) \qquad (3.5)$$

The loss function used to update the weights of the neural network is as follows:

$$Loss = (r + \gamma max_{a'} Q(s',a';\boldsymbol{\theta}) - Q(s,a;\boldsymbol{\theta}))^2 \qquad (3.6)$$

As depicted in Figure 3-2, a standard DQN agent training process consists of two important components, namely the experience relay and the target network. Training large neural networks may lead to divergence, as subsequent updates can be correlated. To address this issue, the experience replay is used, which is operated in the following manner.

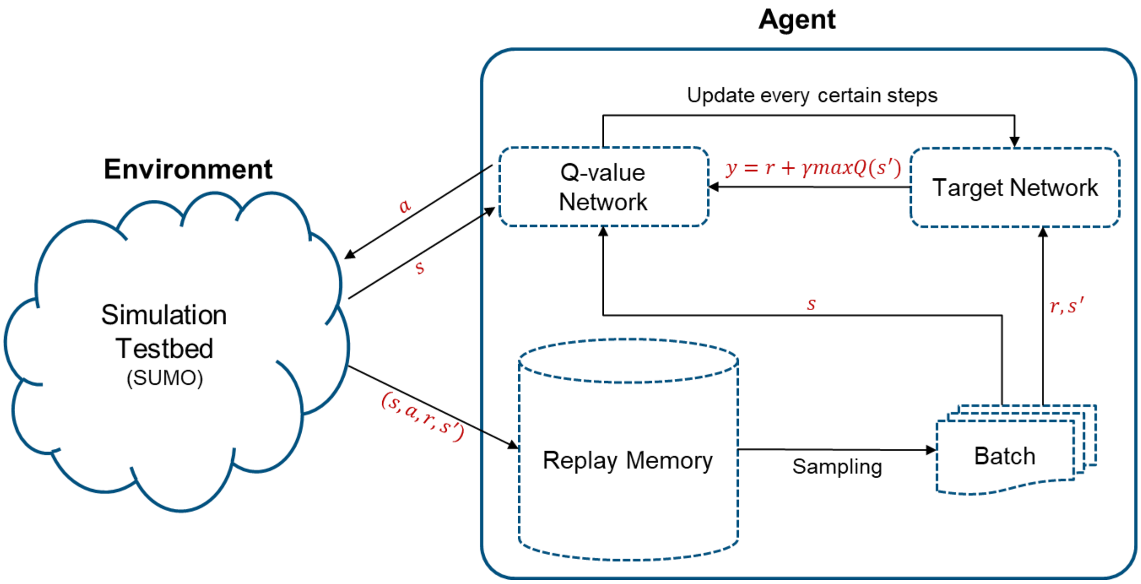


Figure 3-2 The Structure of DQN Used in the Study

- Initialize a memory dataset D.
- Store the experience (s, a, r, s') obtained from the environment for each time step into the dataset.
- Sample a mini-batch of experiences randomly and uniformly from D.
- Train the agent using the mini-batch of experiences instead of the most recent experiences from the environment.
- The memory dataset D stores only a fixed number of recent experiences.

To avoid oscillations or divergence caused by using the same weights θ to calculate both the target value and predicted value in the loss function, a separated network called the target network $\hat{Q}$ is employed to calculate the target value during the training process. Therefore, the loss function is reformulated as follows.

$$Loss = (r + \gamma max_{a'} Q(s', a'; \boldsymbol{\theta}^{-}) - Q(s, a; \boldsymbol{\theta}))^2 \qquad (3.7)$$

The target network works as follows: every C decision step, the weights in the network Q are copied and used to update the target network $\hat{Q}$. The target values y_i for the following C updates are then generated based on the updated target network $\hat{Q}$.

3.1.3. Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a model-free actor-critic DRL algorithm proposed by Schulman et al. (2017). PPO is improved based on Trust Region Policy Optimization (TRPO) introduced by Schulman, Levine, et al. (2015). The actor-critic algorithm has two key components, namely the actor and the critic. The actor, usually refers to as the policy network in DRL, is responsible for selecting actions based on the current state, with the goal of learning an optimal policy. The critic, often refers to as the value network in DRL, evaluates the quality of the action selected by the actor.

The policy can be interpreted as a set of rules used by the agent to choose actions based on the current state. The objective of the actor is to maximize the expected cumulative reward by optimizing the policy. This optimization process can be expressed as follows:

$$\pi^* = arg\max_{\pi} J(\pi) \tag{3.8}$$

where π^* denotes the optimal policy, and the function $J(\pi)$ is used to calculate the expected cumulative reward. This optimization problem is solved by gradient ascent. In DRL, the policy is parameterized by a set of parameters, such as the weight and bias of a neural network. Therefore, it is often expressed as π_θ, where θ refers to the parameter set. The gradient ascent process iteratively updates the parameters θ using the policy gradient $\nabla_\theta J(\pi_\theta)$ with a step size α, which can be expressed as the following equation:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_\theta)|_{\theta_k} \tag{3.9}$$

The most widely used equation to estimate the policy gradient is shown as follows:

$$\nabla_\theta J(\pi_\theta) \approx \widehat{\mathrm{E}_t}[\nabla_\theta \log \pi_\theta (a_t|s_t)\widehat{A_t}] \tag{3.10}$$

where $\widehat{\mathrm{E}_t}$ is the expectation that can be calculated using a batch of samples. $\widehat{A_t}$ is an estimator of the advantage function at timestep t, which evaluates the quality of taking a specific action in a given state compared to the expected average performance. We employ Generalized Advantage Estimation (GAE) to approximate the advantage function, For implementation details of GAE, please refer to the paper of Schulman, Moritz, et al. (2015).

In the implementation of the policy optimization method, a loss function is constructed to facilitate the automatic differentiation process. The loss function is as follows:

$$L^{PG}(\theta) = \widehat{\mathrm{E}_t}[\log \pi_\theta (a_t|s_t)\widehat{A_t}] \tag{3.11}$$

However, in practice, this vanilla policy gradient algorithm may lead to unstable policy updates. PPO is one of the algorithms developed to address this issue, which uses a simple clip operation to constrain the update size. The refined loss function is as follows:

$$L^{CLIP}(\theta) = \widehat{\mathrm{E}_t}[\min(\frac{\pi_{\theta_{new}}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\widehat{A_t}, clip\left(\frac{\pi_{\theta_{new}}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right)\widehat{A_t})] \quad (3.12)$$

where $\pi_{\theta_{new}}(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ denotes the probability ratio between the new policy and the old policy, and ε is a hyperparameter introduced to constrain the update size, usually set to 0.2. In this way, the ratio is constrained within a range determined by ε, therefore limiting the magnitude of policy updates and preventing drastic changes that could lead to instability. Additionally, the loss function is augmented by incorporating the entropy bonus to encourage exploration. Entropy is used to measure the uncertainty or randomness of a policy. Higher entropy values indicate more diverse action selections. The entropy can be calculated using the following equation:

$$H(\pi_\theta(\cdot\,|s_t)) = -\sum_{a\in A}\pi_\theta(a|s_t)\log\pi_\theta(a|s_t) \quad (3.13)$$

Therefore, the loss function utilized for the policy optimization is formulated as follows:

$$L_{actor\_t}(\theta) = \widehat{\mathrm{E}_t}[L_t^{CLIP}(\theta) - cH\big(\pi_\theta(\cdot\,|s_t)\big)] \quad (3.14)$$

where c is the coefficient to adjust the impact of the entropy value. The critic network is denoted as V_ω, with ω as the parameters of the critic network. The goal of the critic network is to minimize the mean squared error, which is given by the following equation:

$$L_{critic\_t}(\omega) = \mathrm{mean}\sum(r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t))^2 \quad (3.15)$$

where $V_\omega(s_t)$ denotes the value based on the state s_t , which is the output of the critic network. γ is the discount factor with a range of $[0, 1]$, and r_t is the immediate reward received from the environment at time step t.

3.1.4. Neural Network Construction

Based on the state space and the DRL algorithm employed in the study, the neural network (NN) consists of three components: the feature extractor, the actor network, and the critic network. The feature extractor takes the observed state information from the environment as input and generates feature vectors. The actor network and the critic network process the output of the feature extractor, generating actions and values, respectively. The detailed architecture of the NN is depicted in Figure 3-3.

In the feature extractor, we employ a convolutional neural network (CNN) similar to what was adopted by Mnih et al. (2015), with slight modifications to accommodate the size of the state space in our study. This CNN consists of five layers, including three convolutional layers, one flatten layer, and one fully connected layer. The first layer has 32 filters with a size of 2*4 and a stride of 1*2. The second layer has 32 filters with a size of 2*3 and a stride of 1*2. The third layer has 32 filters with a size of 2*2 and a stride of 1*1. The output of the fourth layer, after being flattened, is a vector of length 3008. This vector is then processed by a fully connected layer with 128 units, resulting in an output vector of length 128. When the combined state space is used as input, the feature-based vector is concatenated with the output of the fifth layer, producing an output vector of length 144. ReLU activation functions are used in the CNN. The actor network and the critic network have the same architecture, consisting of two fully connected layers with 64 units and a ReLU activation function each, followed by an output layer. When the DQN algorithm is adopted, only the critic network is used to output the Q value. When employing the PPO algorithm, both the actor and critic network are used.

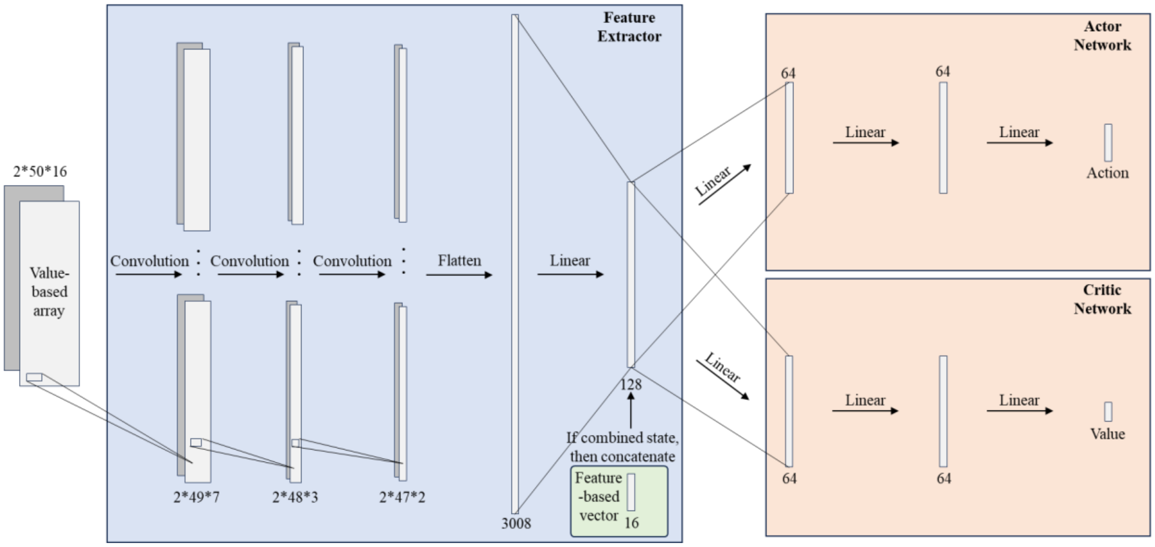


Figure 3-3 Illustration of the Neural Network Structure for Single-Agent PPO

3.2. Multi-Agent Reinforcement Learning

3.2.1. Decentralized Partially Observable Markov Decision Processes Formulation

The traffic signal control problem for multiple intersections can be formulated as a Decentralized Partially Observable Markov Decision Process (DEC-POMDP). In this framework, decentralization involves the utilization of multiple agents, where each agent can only perceive a certain range of the environment and control either a single intersection or a subset of intersections. Within this system, each agent operates according to its individual Partially Observable Markov Decision Process (POMDP) and interacts with each other. A DEC-

POMDP can be defined as a tuple $\langle S, A, O, R, P, n, \gamma \rangle$. S is the state space, represents the set of possible states in the system. A is the action space, $A = \{a_i, \dots, a_n\}$ is the joint action of all the agents. O is the local observation space, $o_i \ = \ O(s; i)$ denotes the partially observed information specific to agent i. $R(s,\ A)$ defines the shared reward function, which calculates the feedback according to the current state s and the joint action A. $P(s'|s,\ A)$ is the transition probability from s to s' given the joint action A. n is the number of agents involving the DEC-POMDP. γ is the discount factor, which functions similarly to the discount factor in an MDP.

3.2.1.1. Local observation

The local observation is a vehicle-based array state, which is the same as the setting in the isolated intersection scenario in section 3.1.1.1. In the DEC-POMDP framework, each agent i obtain its local observation $o_{i,t}$ at decision time step t.

3.2.1.2. Global state

The global state has two components and can be denoted as $S_t = \{o_t, p_t\}$. o_t is the set of the local observations from all the agents at decision step t, which can be expressed as $o_t = \{o_{1,t}, \dots, o_{n,t}\}$. p_t is the set of the phase state of all the agents at decision step t, which can be expressed as $p_t = \{p_{1,t}, \dots, p_{n,t}\}$. Each phase state is represented using the one-hot encoding technique, forming a vector with a length equal to the number of phases plus a yellow phase and an all-red phase. For example, in a typical four-leg intersection with eight phases, the encoded vector would have a length of ten.

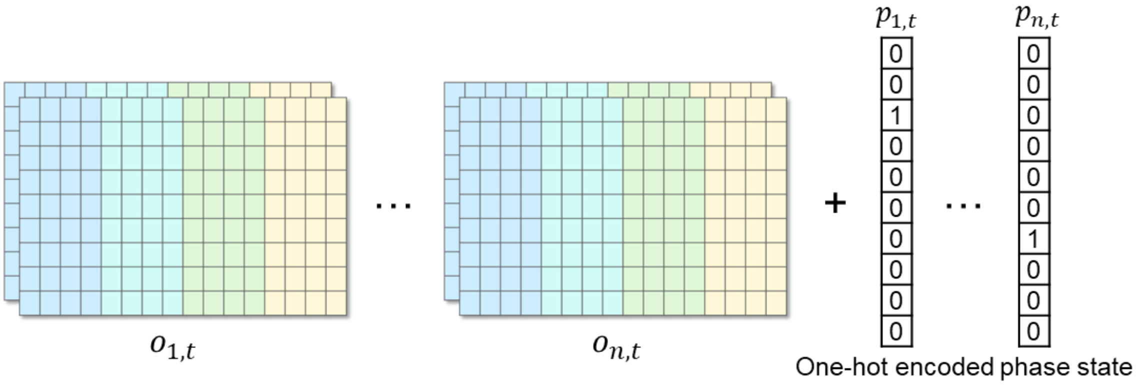


Figure 3-4 Illustration of Global State Representations for Corridor in the Study

3.2.1.3. Action space

Two types of action spaces are utilized in the study, namely discrete action space and multi-discrete action space. The settings of these action spaces are the same as defined in the isolated intersection scenario in section 3.1.1.2. Each agent i selects its action $a_{i,t}$ at decision step t.

3.2.1.4. Reward function

In multi-agent scenarios, the reward function for each agent is still the difference in the cumulative person delay, the calculation process also follows the definition in section 3.1.1.3.

3.2.2. Multi-Agent Actor Critic

Traditionally, MARL can be implemented in two frameworks, i.e., centralized and decentralized. In centralized implementation, a single policy is trained to generate joint actions for all agents. However, this framework may face scalability challenges. On the other hand, decentralized implementation involves each policy optimizing its own reward independently. While it can address scalability problems, it may suffer from instability issues due to the non-stationary problem.

We follow the study conducted by Yu et al. (2022), utilizing PPO as the training algorithm and employing the centralized training and decentralized execution (CTDE) framework in this study, as shown in Figure 3-5. In the CTDE framework, global observations are used as input for the centralized critic network during the training stage, outputting a more accuracy critic values, and therefore providing more precise guidance for the gradient update of the actor network. This strategy effectively mitigates the non-stationary issues. During the execution stage, only local observations are needed to generate actions for each agent, providing a robust solution to scalability concerns. The gradient update mechanism in this framework can be expressed mathematically as follows (Lowe et al., 2017).

$$\nabla_{\theta_i} J\left(\pi_{\theta_i}\right) \approx \mathrm{E}[\nabla_{\theta_i} \log \pi_i\,(a_i|o_i)\mathrm{Q}_i^{\pi}(\mathrm{s}, a_1, \dots, a_n)] \tag{3.16}$$

The above equation is derived from the classical policy gradient equation employed in policy-based RL. However, a key difference lies in the computation of Q values, where inputs are the global state s and the joint actions of all agents, $a_1, \dots, a_n$. Consequently, the Q value function takes the form of a centralized function $\mathrm{Q}_i^{\pi}(\mathrm{s}, a_1, \dots, a_n)$. In this study, we opt for the advantage function and employ the GAE method to approximate advantages.

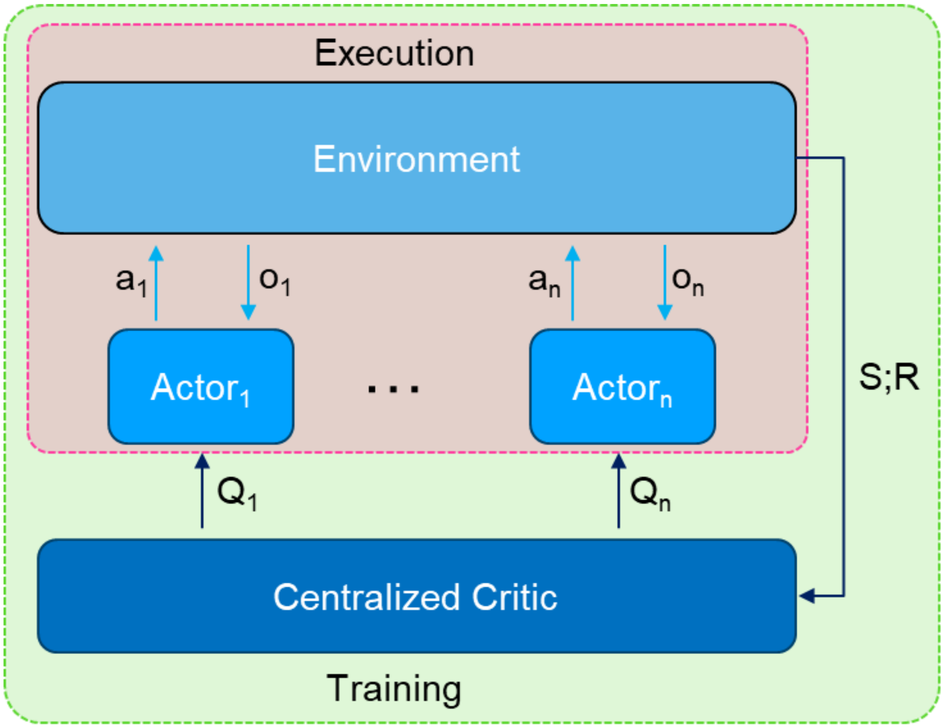


Figure 3-5 Illustration of Centralized Critic and Decentralized Actor

3.2.3. Neural Network Construction

The neural network structure employed in this study is shown in Figure 3-6. The are also three major components, namely feature extractor, actor network, and critic network. The critic network receives a concatenated vector that incorporates the outputs of feature extractors along with the phase states of all agents. It then computes advantage values for each actor, forming a centralized critic network. This centralized critic network assesses the effectiveness of actions and guides the optimization process of the actor network. On the other hand, the actor network takes as input the output vector from the corresponding feature extractor of an individual agent. The output of the actor network is a vector containing probabilities for each possible action for the specific agent. Operating in a decentralized manner, it focuses solely on the local observations of each agent.

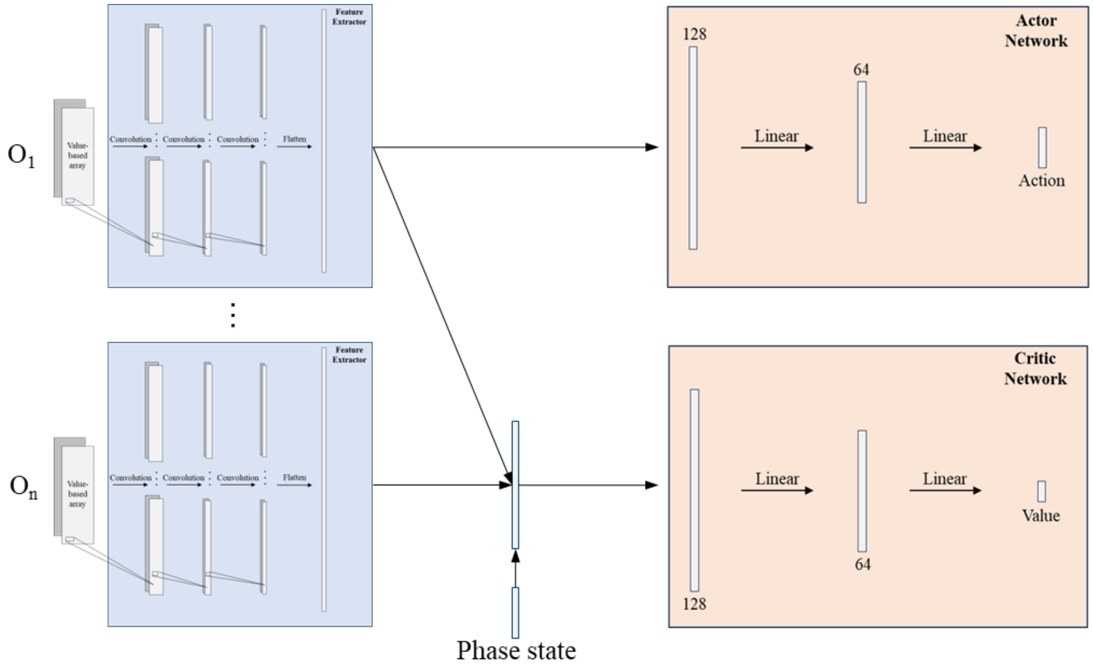


Figure 3-6 Illustration of the Neural Network Structure for Multi-Agent PPO

Chapter 4. Isolated Intersection

This chapter focuses on the experimental settings and result analyses regarding isolated intersection scenarios.

4.1. Traffic Configuration

In order to evaluate the performance of the proposed traffic signal controllers, a simulation testbed is built using Simulation of Urban MObility (SUMO), an open-source traffic simulation software that can be controlled via the Traffic Control Interface (TraCI) by Python. A typical four-approach intersection of Central Avenue and Eastway Drive in Charlotte, North Carolina, U.S.A. is selected as the test intersection, as shown in Figure 4-1. Each approach has 4 lanes. In the north and southbound approaches, there are two lanes for through traffic and one exclusive lane each for left-turn and right-turn traffic, respectively. In the east and westbound approaches, there are two dedicated left-turn lanes, one through lane, and one right-turn and through shared lane. The yellow time is set to 3 seconds and the red clearance time is 2 seconds. The speed limit for the south-north road is 45 mph and for the east-west road is 35 mph. Buses operate on a north-south route only.

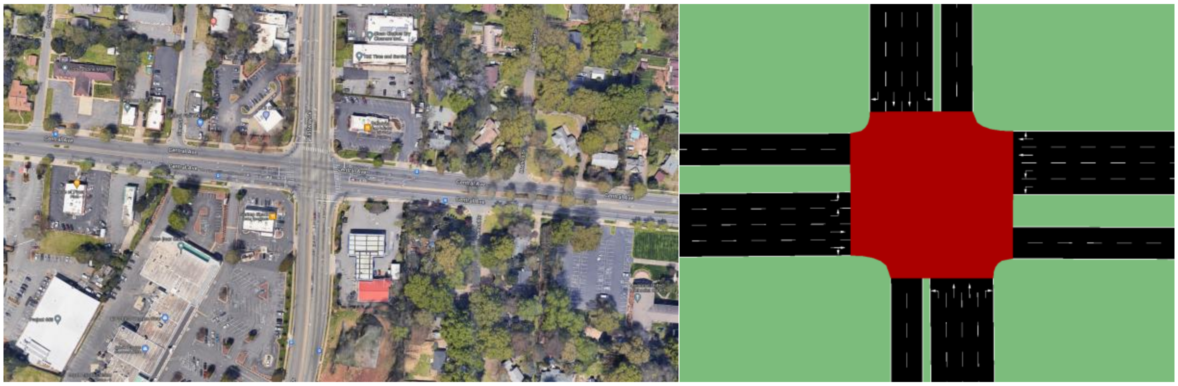


Figure 4-1 Layout for the Test Isolated Intersection

Table 4-1 Traffic Volume of Each Travel Direction, veh/h

Time	SB			WB			NB			EB		
period	R	T	L	R	T	L	R	T	L	R	T	L
PM Peak	176	793	88	68	341	206	325	883	180	193	547	246
Off-peak	152	707	36	40	319	235	122	541	138	128	197	91

Note: SB=southbound; WB=westbound; NB=northbound; EB=eastbound; R=right turn; T=Through, L=Left turn.

4.2. Simulation Settings

The intelligent driving model (Treiber et al., 2000) and LC2013 Model (Erdmann, 2015) are employed to control the longitudinal and lateral movements of the vehicle, respectively. The car-following and lane-changing parameters are the same for both connected and unconnected vehicles. For traffic demand, the peak hour and off-peak hour volumes at the test intersection are used. The peak hour is 5 - 6 PM on Wednesday, April 21, 2021, and the off-peak hour is 9 - 10 AM that day. The traffic volume data is obtained from the Charlotte Department of Transportation (CDOT) and is presented Table 4-1. The traffic flows generated in the simulation follow the Poisson distribution. Each simulation run lasts one hour, with a ten-minute warm-up period.

Six traffic signal controllers are developed based on corresponding signal control strategies. Specific scenarios are established based on these basic simulation environments by considering factors such as traffic demand, bus occupancy, CV market penetration rate (MPR), and bus arrival headway. For simplicity, the passenger occupancy of cars is set to be one passenger per vehicle. The passenger occupancy of buses is also set as a constant, but it varies according to the specific scenario settings. The basic simulation environment conditions and the corresponding signal control strategies are detailed as follows.

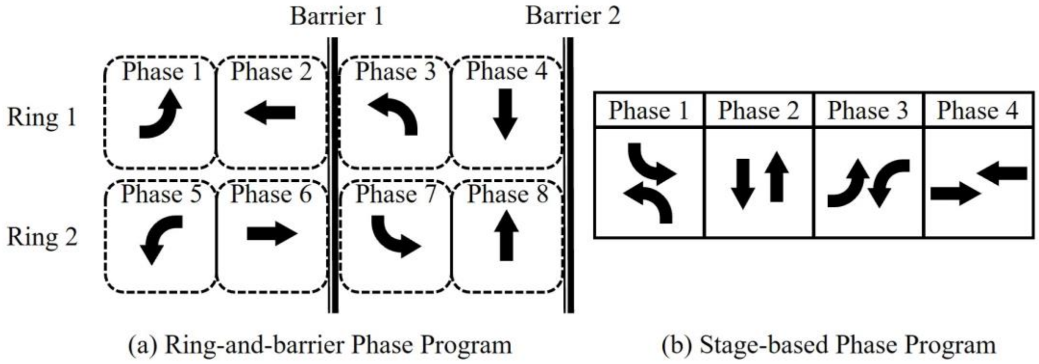


Figure 4-2 Signal Phase Program Used in the Research

- **Pretimed Signal Controller (PSC):** In this controller, a stage-based phase program is used, as shown in Figure 4-2(b). The signal timing for each phase is calculated using the Webster method (Koonce, 2008). Both buses and cars are human-driven vehicles (HDVs).

- **Actuated Signal Controller (ASC):** A fully actuated signal control strategy is adopted in this controller. A typical National Electrical Manufacturers Association (NEMA) phase diagram is adopted, and the phase sequence is shown in Figure 4-2(a). The minimum and maximum green time are set following the signal plan obtained from the Charlotte Department of Transportation. Both buses and cars are HDVs.

- **Actuated Signal Controller with TSP Using the Traditional Detector (ATSP-T):** In this controller, the bus detectors are placed 100 meters before the stop line in the south and north approaches. When a bus crosses the detector, the signal will be switched to the corresponding phase. Otherwise, the SPaT is controlled by fully actuated control strategy. The buses and cars are HDVs.

- **Actuated Signal Controller with TSP Using CV (ATSP):** Buses are CVs, and no bus detector is installed. When a bus approaches the intersection within 100 meters, the signal will be switched to the corresponding phase. Otherwise, the SPaT is controlled by fully actuated control strategy. The cars are HDVs.

- **GA Optimized Signal Controller with TSP (GA-TSP):** The stage-based signal phase shown in Figure 4-2(b) is adopted in the GA optimizer. The decision variable is the duration of green time for each phase. The minimum green time for the left turn phases is 6 seconds and for the through phases it is 12 seconds. The maximum green time is 20 seconds for the left turn phases and 35 seconds for the through phases. Accordingly, the cycle length ranges from 56 seconds to 130 seconds. Buses are CVs, and the MPR of cars varies from 20% to 100% in 20% intervals. For the parameters related to the genetic algorithm, the maximum generation is set to 250, the population size is 20, the probability of mutation is 0.7, and the probability of crossover is 0.7. Elitism is applied to retain the best solution in a generation.

- **DQN signal controller with TSP (DQN-TSP)**: In this controller, the SPaT is controlled by a DQN agent. All vehicles, including both buses and cars, are CVs, and hence, their real-time positions and speeds are available. The communication range between CVs and the DQN agent is set to 350 meters. Note that in scenarios where the MPR is below 100%, the states of unconnected vehicles are not considered by our controller.

In addition, we delve into the problem of robustness enhancement of the DRL-based traffic signal controllers in mixed traffic environments. To this end, we have further developed four DRL-based signal controllers, outlined as follows:

- **Double DQN Signal Controller (DDQNSC):** A DRL agent is implemented to control the traffic signal. The control algorithm employed in the agent is Double DQN (DDQN), which is known for its improved stability compared to the vanilla DQN algorithm. For detailed implementation of DDQN, please refer to the papers written by Mnih et al. (2015) and Van Hasselt et al. (2016). In this agent, the state space is the vehicle-based array state, the action space is the discrete action space, and the reward function is defined as described in section 3.2.1.1.

- **PPO Signal Controller (PPOSC):** PPOSC utilizes the same state space, action space, and reward function as in DDQNSC, but the control algorithm is PPO.

- **PPO Signal Controller with Multi-discrete Action (PPOSC-M):** PPOSC-M utilizes the multi-discrete action space. Other **than** that, the state space, reward function, and control algorithm are the same as in PPOSC.

- **PPO Signal Controller with Multi-discrete Action and Combined State (PPOSC-M-C):** PPOSC-M-C utilizes both the multi-discrete action space and the combined state space. The reward function and the control algorithm remain the same as in PPOSC.

4.3. Training

The DQN-TSP agent's experience is stored in a replay memory, with a capacity of 50,000 experiences, following a First-in-First-Out storage rule. The discount factor for the agent is set to 0.65, and the batch size for updating the model is set to 32. The training process employs the Adam optimizer with a learning rate of 0.001. The action selection process follows an ε-greedy policy, where ε decreases as the number of episodes increases. The equation used to determine ε is presented below.

$$\varepsilon = 0.01 + (0.9 - 0.01) * exp(-0.1 * episode) \tag{4.1}$$

The DQN-TSP agent employs a neural network consisting of three convolutional layers and three fully connected layers, as outlined in Table 4-2. The input obtained from the simulation testbed is an image-like representation with dimensions of 50*16*2, and the output is 8 actions representing 8 possible phases.

Table 4-2 The Neural Network Structure of DQN-TSP

Layer	Filter size	Stride	Num Filters	Activation
Conv1	2*4	1*2	32	ReLU
Conv2	2*3	1*2	32	ReLU
Conv3	2*2	1*1	32	ReLU
Fc4			128	ReLU
Fc5			64	ReLU
Fc6			8	Linear

The simulations are run on a laptop equipped with an AMD Ryzen 7 5800H processor, 32 GB of RAM, an NVIDIA GeForce RTX 3070 Laptop GPU, and the Windows 10 operating system. Two DQN-TSP agents are trained, one based on the peak hour traffic demand, and the other based on the off-peak hour traffic demand. The training time required for the peak agent and the off-peak agent is 6 hours and 3 hours, respectively. The training curves are shown in Figure 4-3.

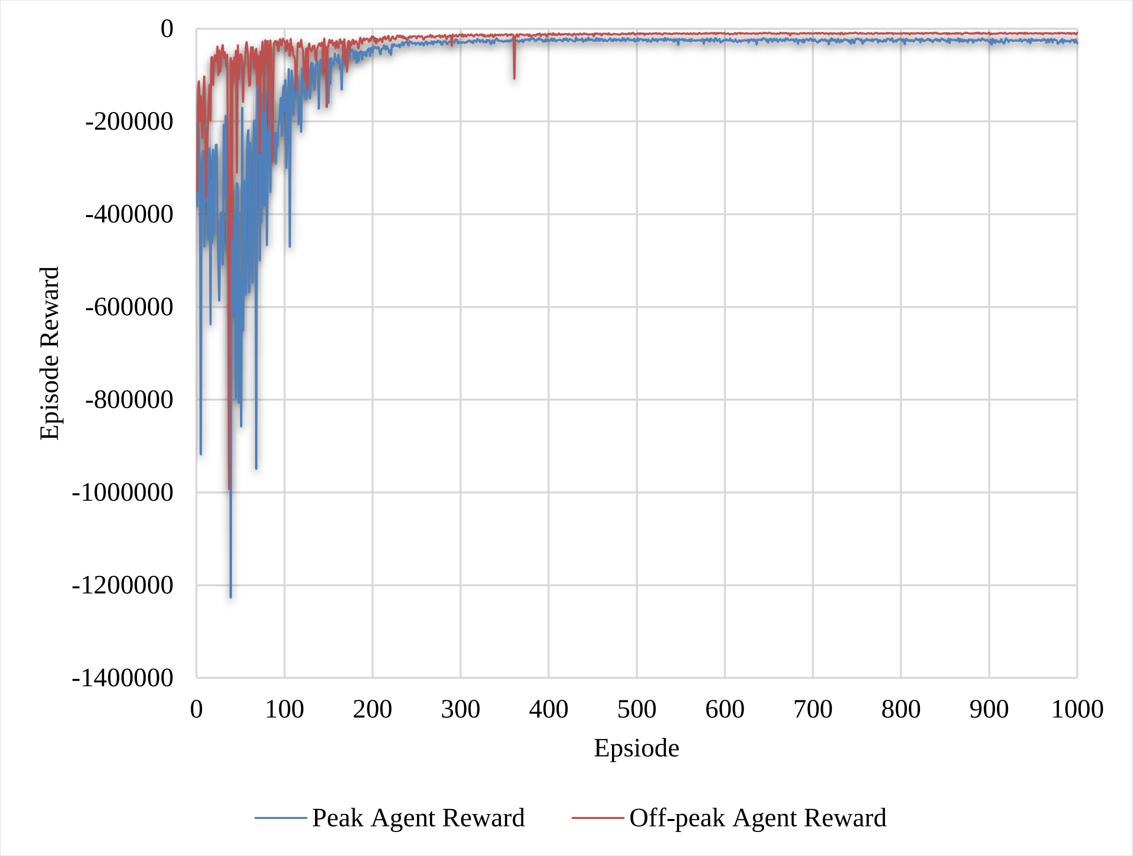


Figure 4-3 Episode Reward Curves for Peak and Off-peak DQN-TSP Agents

These DDQSC, PPOSC, PPOSC-M, PPOSC-M-C are also trained before evaluation to ensure stable performance. Specifically, they have been trained using both peak and off-peak demand scenarios with 100% MPR, generating a total of eight trained controllers. Python libraries used to implement these controllers include TracI, Gymnasium, Pytorch, and Stable-baseline3. The Adam (adaptive moment estimation) optimizer is employed in the training process. The hyperparameters of both DDQN and PPO have been well-tuned, and their values are presented in Table 4-3.

Table 4-3 Hyperparameters Used for DDQN and PPO

Hyperparameter	DDQN	PPO
Training steps	400,000	400,000
Discount factor	0.65	0.65
Learning rate	0.0003	0.0001 (actor), 0.00005 (critic)
Buffer size	50,000	-
Batch size	32	64

Target update interval	200	-
Exploration rate	Decrease from 1 to 0.01	-
Clip range	-	0.2
Entropy coefficient	-	0.01
Max gradient update	-	0.5

Their training performance under both peak and off-peak demand with 100% MPR is shown in Figure 4-4. While all controllers can converge to similar rewards, the PPO algorithm exhibits faster convergence and more stable performance compared to the DDQN algorithm, particularly in peak traffic conditions. The utilization of the multi-discrete action space effectively accelerates the training process. Additionally, employing both the multi-discrete action space and the combined state space ensures a more stable training performance.

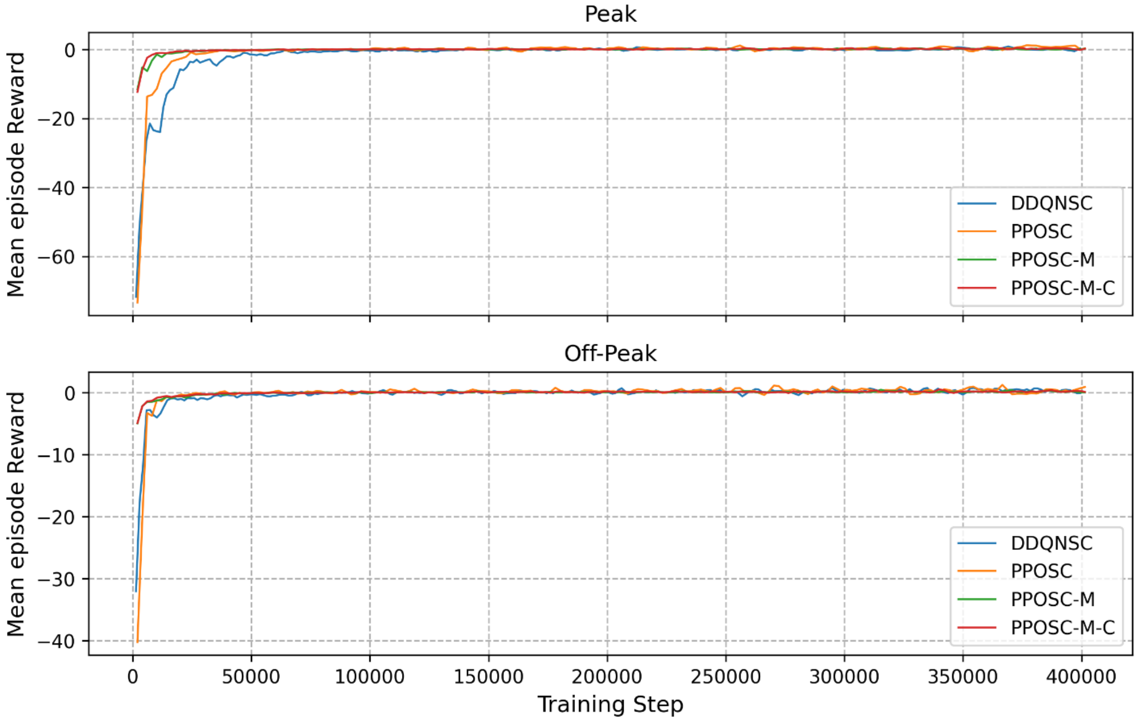


Figure 4-4 Mean Episode Reward Curves for DRL-based TSC Controllers under Peak and Off-peak Conditions

4.4. Result Analyses

The experimental results are presented in two sections. The first section focuses on the performance of the DRL-based signal controllers concerning TSP strategy. The second section focuses on evaluating the robustness of DRL-based signal controllers in mixed traffic environments, without considering TSP.

4.4.1. TSP Performance Evaluation

The performance metrics used to evaluate traffic performance are average bus delay, average car delay, and average person delay. Each scenario is run for a simulation time of one hour, with a warm-up period of ten minutes that is excluded from the analysis of the results. To ensure a more accurate evaluation of performance, the metrics for each scenario are averaged over fifty runs with different seeds. In addition, the random seeds are kept consistent across scenarios to guarantee a fair comparison. The DQN-TSP agent's performance is evaluated using peak and off-peak agents, respectively, for peak and off-peak traffic demands.

4.4.1.1. Performance Evaluation

The performance of six basic scenarios with different signal control strategies is evaluated and compared, with the performance of PSC serving as the baseline. In these scenarios, each bus is set to have a passenger occupancy of 30 passengers, while the number of passengers in each car is set to 1. The average bus arrival headway is set to five minutes. A detailed comparison of these six basic scenarios is shown in Table 4-4.

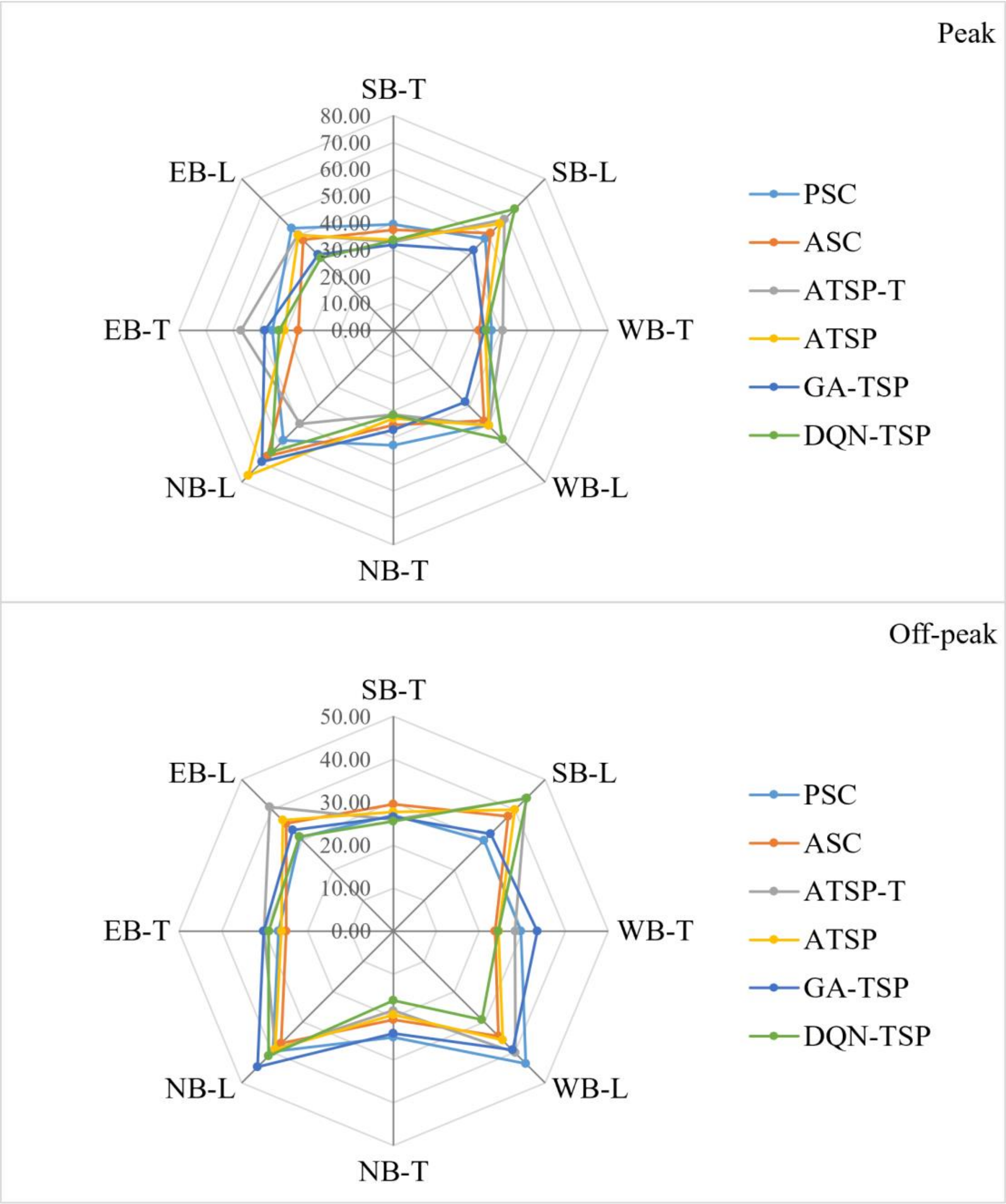
Table 4-4 Comparison of Average Delay for Basic Scenarios at Isolated Intersection Considering TSP

Demand	Type	PSC	ASC	ATSP-T	ATSP	GA-TSP	DQN-TSP
Peak	Average Bus Delay (s)	41.43	38.67	18.61	16.94	30.75	19.43
	Delay Change		-6.66%	-55.08%	-59.11%	-25.78%	-53.10%
	Average Car Delay (s)	40.19	35.48	38.02	36.14	36.27	35.24
	Delay Change		-11.72%	-5.40%	-10.08%	-9.75%	-12.32%
	Average Person Delay (s)	40.38	35.97	35.96	33.18	35.68	32.80
	Delay Change		-10.92%	-10.93%	-17.83%	-11.63%	-18.77%
Off-peak	Average Bus Delay (s)	28.52	27.80	13.73	12.37	22.13	13.10
	Delay Change		-2.52%	-51.86%	-56.63%	-22.41%	-54.07%
	Average Car Delay (s)	27.31	25.49	26.00	25.27	27.94	23.32
	Delay Change		-6.66%	-4.80%	-7.47%	2.31%	-14.61%
	Average Person Delay (s)	27.57	25.99	24.16	22.50	27.07	21.13
	Delay Change		-5.73%	-12.38%	-18.39%	-1.82%	-23.36%

The DQN-TSP has the best performance in terms of average person delay under both peak and off-peak traffic demand conditions. Compared to the baseline, the proposed DQN-TSP controller reduces average person delay by 18.77% and 23.36% in peak and off-peak conditions, respectively. The larger reduction in the off-peak condition suggests that there may be more room for improvement in traffic efficiency under lower traffic demand conditions. The ATSP has the lowest average bus delay in both peak and off-peak conditions, reducing by 59.11% and 56.63% respectively compared to the baseline. However, in the peak condition, the average car delay of ATSP is 36.14 seconds, slightly higher compared to both ASC and DQN-TSP controllers. It is due to the unconditional priority given to buses in ATSP. Yet, during off-peak, ATSP's average car delay of 25.27 seconds is slightly lower than ASC's, implying that granting priority to buses has a less negative impact on other traffic in low-traffic demand conditions. In addition, the comparison of the two actuated control strategies with TSP indicates that the CV technology offers a better performance than just using traditional fixed detectors to sense bus arrivals. As for the TSP-GA scenario, the average bus delay decreases by 25.78% and the average car delay decreased by 9.75% during the peak hour. The average bus delay is reduced by 22.41% and the average car delay increases by 2.31% during the off-peak hour. These results indicate that the GA optimizer with TSP performs better in peak hours than in off-peak hours.

The detailed impacts of the six basic control strategies on average vehicle delays in each traffic movement direction are shown in Figure 4-5. Right-turn movements are not presented as signal control strategies have little impact on these directions. In peak traffic demand, PSC and ASC provide balanced services in all directions. Meanwhile, those TSP controllers need to grant priority to buses, resulting in longer waiting times for vehicles in conflicting movements, such as southbound and northbound left turns. Such adverse impacts are mitigated during off-peak, which makes sense. However, compared to ATPS-T and ATSP, DQN-TSP can provide more balanced services to vehicles in conflicting directions. For example, in the ATSP scenario, the average vehicle delay in southbound and northbound left turns is 56.01 and 76.56 seconds, respectively. Meanwhile, in the DQN-TSP scenario, the average vehicle delay in both directions is about 64 seconds. In the off-peak traffic demand condition, all three scenarios, except for PSC and GA-TSP, maintain similar performance in all directions as there is no need to sacrifice other directions to prioritize buses. The unbalanced performance in PSC and GA-TSP during off-peak is due to an imbalance in traffic volume, with westbound left-turn having 2.6 times the volume of eastbound left-turn, but they still share the same phase. Compared to PSC, during the peak hour, average vehicle delays of GA-TSP are reduced by 12-25% in almost all left turn directions, except for a 19.62% increase in the northbound left turn. This is understandable, as the traffic demand for northbound left turn is more than twice that of southbound left turn. Regarding the through traffic, the average vehicle delay is reduced by 19.39% and 13.53% in southbound and northbound, respectively. During the off-peak hour, average vehicle delays for GA-TSP decrease in southbound through, westbound left, and northbound through. The average vehicle delays of other travel directions increased by ranging from about 7% to 13%.

These results indicate that GA-TSP and DQN-TSP have the potential to provide conditional priority to buses while minimizing the negative impact on conflicting traffics. However, the DQN controller outperforms the GA controller in all metrics.



Note: SB=Southbound, WB=Westbound, NB=Northbound, EB=Eastbound, T=Through, L=Left turn.

Figure 4-5 Average Vehicle Delay in Each Direction of Basic Scenario Considering TSP

4.4.1.2. Sensitivity Analysis

- CV Market Penetration Rate

In the basic scenario, we assume that both cars and buses are CVs, so all the real-time traffic information of vehicles is available. However, as CV technology is still in its early stages of development, it will take many years for a fully CV environment to become a reality. Furthermore, due to privacy concerns, 100% market penetration of CVs may never be reached. Thus, it is crucial to investigate the impact of the CV market penetration rate on the performance of the proposed controllers. Ten scenarios have been designed, covering both peak and off-peak traffic demand conditions, with MPR ranging from 20% to 100% in increments of 20%. Other settings are the same as in the basic scenarios.

The results shown in Figure 4-6 illustrate that, with the increase in MPR, the performance of the proposed controllers improves across all metrics and scenarios. During the peak hour, the average bus delay is lower than the baseline, even with the MPR being as low as 20%. In off-peak hours, the DQN-TSP controlled average bus delay is lower than the baseline at an MPR of 40%, whereas the GA-TSP controlled average bus delay outperforms the baseline at an MPR of 20%. This suggests that DQN-TSP requires a certain threshold of information to ensure satisfactory performance, while the performance of GA-TSP is more robust than DQN-TSP in low MPR environments. These results are consistent with findings from the broader field of RL research, indicating the partially observable issue is a research topic worthy of attention. When controlled by DQN-TSP, during peak hours, the average person delay is lower than the baseline when the MPR exceeds 60%. Furthermore, even with an MPR as low as nearly 40% in off-peak hours, the performance in terms of average person delay is better than the baseline. These results suggest that the proposed DQN controller also has a certain level of robustness even when only partial traffic information is available.

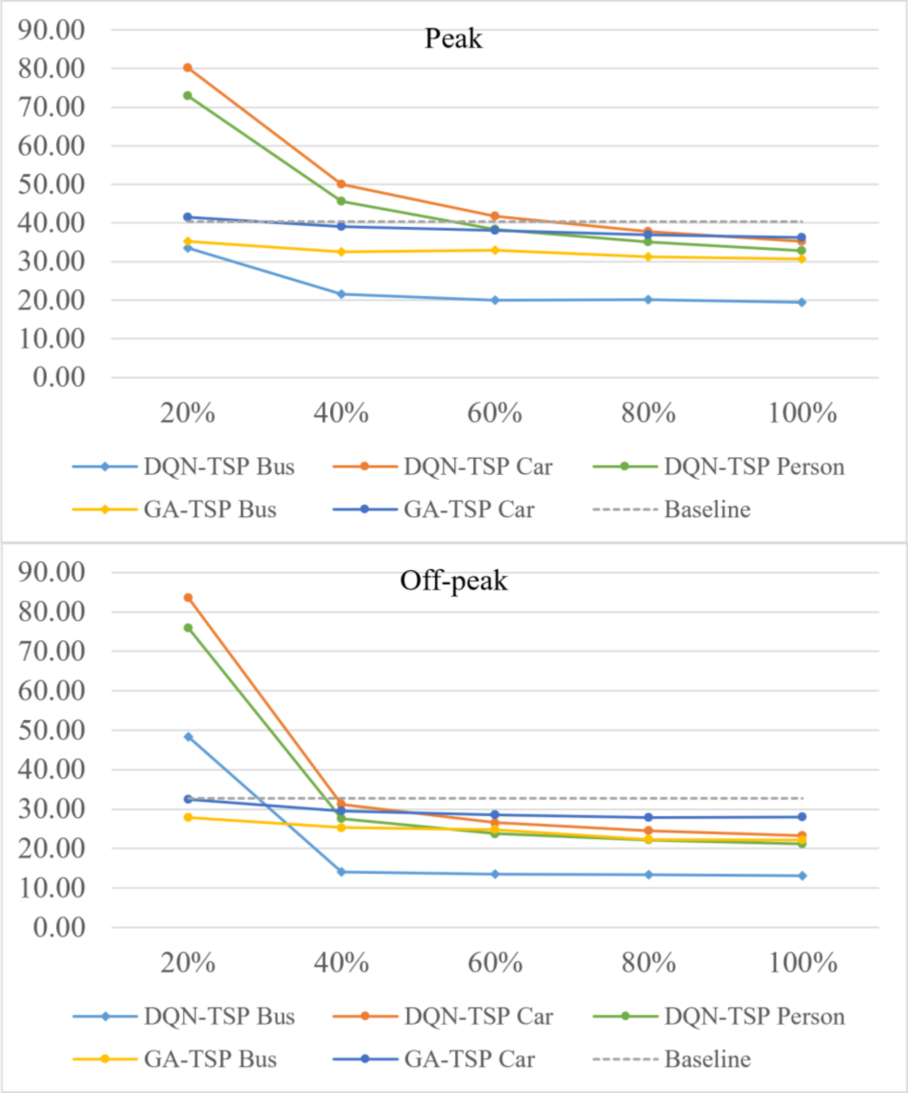


Figure 4-6 Sensitivity of Controllers to CV Market Penetration Rate at Isolated Intersection

- Bus Passenger Occupancy

We all know in the real world, the passenger occupancy of buses, which can be easily obtained through CV technology, varies from bus to bus. To study the sensitivity of the proposed controllers to changes in bus occupancy, we conducted experiments where the bus occupancy is varied while all other settings remain the same as in the basic scenarios. In this sensitivity analysis, the number of passengers on each bus is different in each scenario, including 1 passenger per bus, 10 passengers per bus, 30 passengers per bus, 50 passengers per bus, and 70 passengers per bus. The results shown in Figure 4-7 indicate that, as the bus occupancy increases, both the average bus delay and average person delay decrease during both peak and off-peak hours. At the same time, the average car delay experiences a slight increase. Additionally, even as bus occupancy continues to increase, the increase in average car delays does not accelerate, while the decline in average bus delays becomes more moderate. In addition, compared to DQN-TSP, the GA-TSP controllers exhibit lower sensitivity. When the bus occupancy is greater than 10 passengers per vehicle, the average bus delays for GA-TSP largely remain constant, especially during peak hours. These findings underscore the superior capability of the proposed DQN-TSP controllers in handling fluctuations in bus occupancy, a crucial feature that enhances their suitability for real-world application.

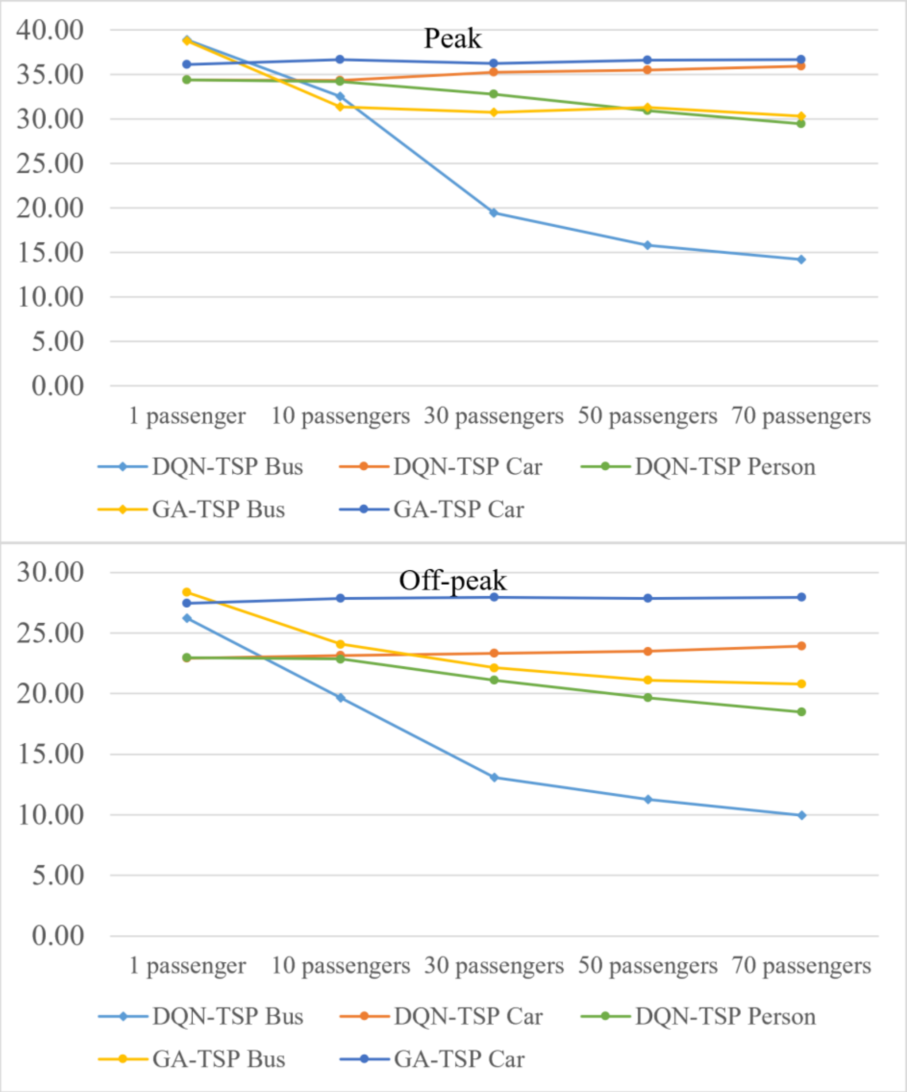


Figure 4-7 Sensitivity of Controllers to Bus Occupancy at Isolated Intersection

- Bus Arrival Headway

In this part, the impact of bus arrival headway on the effectiveness of the proposed controllers is explored by considering five different headways, including 2 minutes, 5 minutes, 10 minutes, 15 minutes, and 30 minutes, under two different traffic demand conditions. The rest of the scenario settings are in line with the basic scenarios. As illustrated in Figure 4-8, an increase in bus arrival headway results in a decrease in the average car delay and an increase in the average person delay. Meanwhile, the average bus delay also decreases a little. This aligns with expectations, as with the increase in headway, fewer buses are traveling on the road, allowing the traffic signal to give more consideration to the cars. As the headway increases, the gap between the average person delay and average car delay becomes closer. However, the impact of changes in bus arrival headway on traffic performance is not significant.

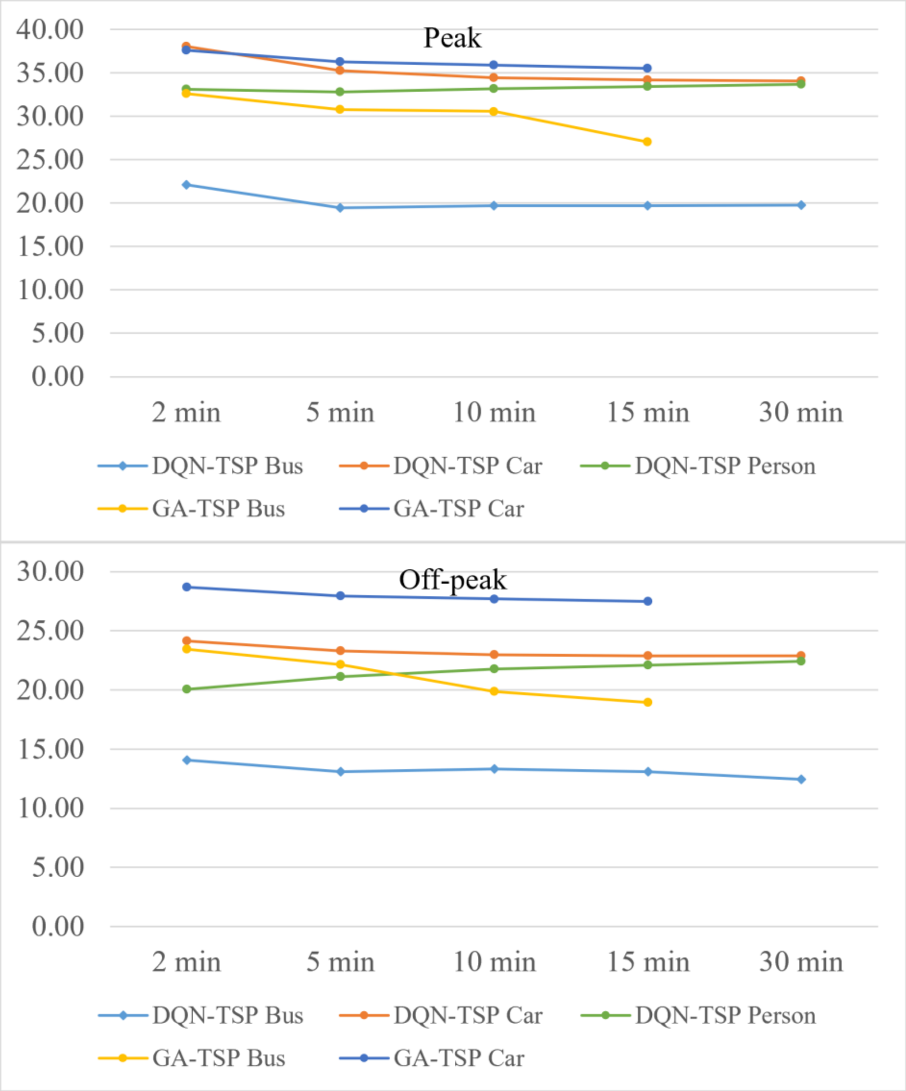


Figure 4-8 Sensitivity of Controllers to Bus Arrival Headway at Isolated Intersection

4.4.2. Robustness Evaluation

In this section, we focus on evaluating the robustness performance of DRL-based signal controllers in mixed traffic environments without considering TSP. We have developed four DRL-based controllers, i.e., DDQNSC, PPOSC, PPOSC-M, and PPOSC-M-C. Their settings are detailed in previous sections.

4.4.2.1. Comparison of Basic Scenarios

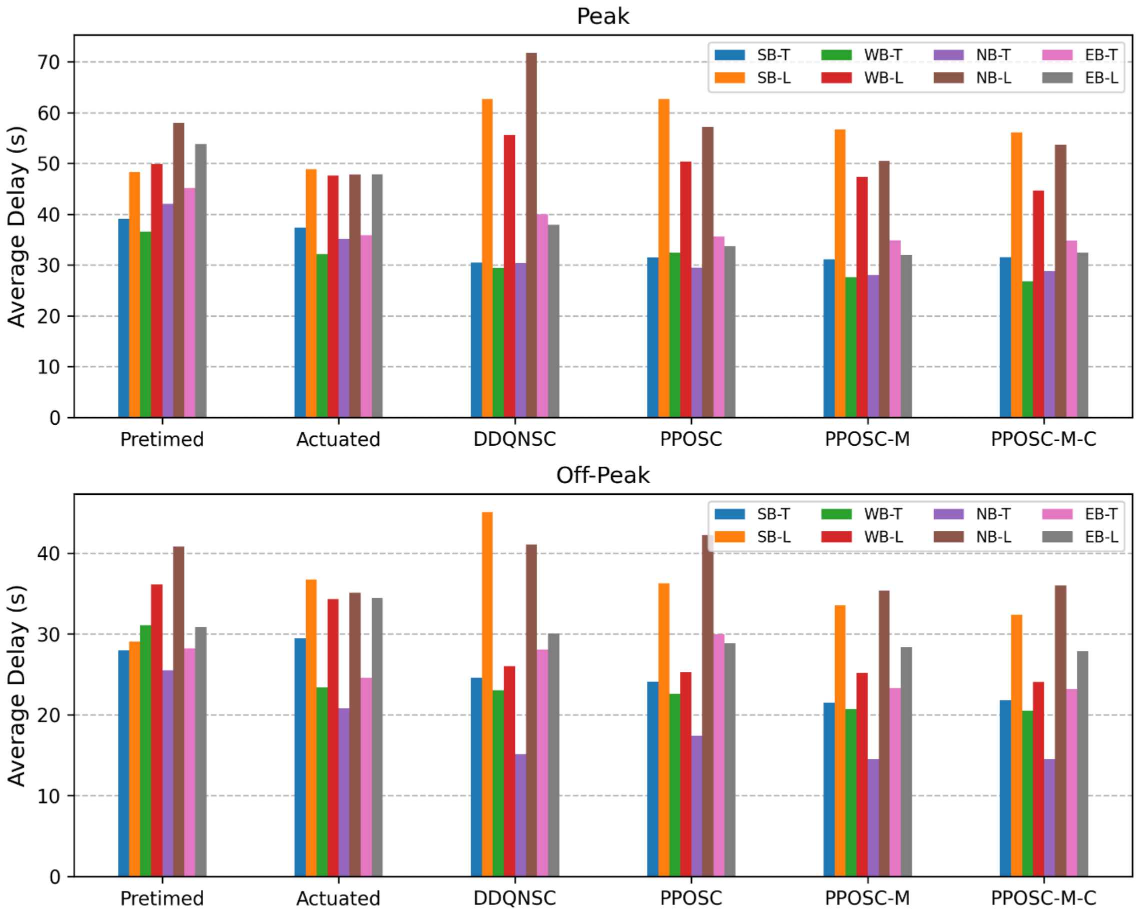
Firstly, we compare traffic performance in basic scenarios in terms of average vehicle delay. All settings remain the same except for differences in traffic demand and control strategies. The performance of PSC is used as the baseline for comparison. As shown Table 4-5, all four DRL-based controllers outperform these two traditional controllers under both peak and off-peak conditions in terms of the average delay. PPO-based controllers perform slightly better than DDQNSC. PPOSC-M shows the best performance in the peak scenario, with a 24.29% reduction in average delay compared to the baseline. PPOSC-M-C has the best performance in the off-peak scenario, with a 27.24% reduction of average delay compared to the baseline. The standard deviations of the average delay among 50 simulation runs demonstrate that PPOSC-M has the most stable performance in the off-peak scenario. In peak scenarios, PSC provides the most stable service while PPO-based controllers show almost the same level of stable performance. These results indicate that PPO-based controllers, especially PPOSC-M and PPOSC-M-C, can provide better service while ensuring good stability.

Table 4-5 Performance Comparison of Basic Scenarios without TSP

Type	Peak					
	PSC	ASC	DDQNSC	PPOSC	PPOSC-M	PPOSC-M-C
Average Delay (s)	39.90	34.58	33.47	31.90	30.21	30.37
SD	0.80	0.95	1.38	0.97	1.06	0.99
Delay Change		-13.34%	-16.12%	-20.06%	-24.29%	-23.89%
Type	Off-peak					
	PSC	ASC	DDQNSC	PPOSC	PPOSC-M	PPOSC-M-C
Average Delay (s)	27.35	25.21	22.24	22.49	19.97	19.90
SD	0.65	0.73	0.79	0.63	0.56	0.61
Delay Change		-7.84%	-18.69%	-17.77%	-27.00%	-27.24%

As presented in Table 4-1, the traffic volumes in the real world are unbalanced, such as the northbound left-turn volume being more than 2 times higher than the southbound left-turn volume during the peak hour. To evaluate the effectiveness of traffic signal controllers in a more comprehensive manner, it is necessary to investigate vehicle delays in different turning movements. Figure 4-9 illustrates the average delay of each turning movement under basic scenarios for each controller. Right-turn Delays are excluded from the comparison as they have little to do with the controller’s behavior. In general, left-turning vehicles experience longer delays compared to those traveling through. In terms of service balance across all directions, the traditional controllers, PSC and ASC, outperform the DRL-based controllers. In peak scenarios, DDQNSC exhibits the least balanced service, with a maximum delay difference of 42.33 seconds between the northbound left-turn (71.73 s) and the westbound through (29.40 s). Similarly, in off-peak scenarios, DDQNSC also provides the least balanced service, with a maximum delay difference of 29.91 seconds between the southbound left-turn (45.07 s) and the northbound through (15.16 s). However, PPO-based controllers, especially those using multi-discrete action space, namely PPOSC-M and PPOSC-M-C, can provide nearly the same level of balancing services as traditional controllers.

Based on these comparisons, PPO-based controllers using multi-discrete action space have the most favorable performance. They not only demonstrate the most effective performance in terms of overall average delay but also exhibit a good balance in serving vehicles traveling in different directions.



Note: SB=Southbound, WB=Westbound, NB=Northbound, EB=Eastbound, T=Through, L=Left turn.

Figure 4-9 Average Delays in Each Turning Movement of Basic Scenarios without TSP

4.4.2.2. Comparison of Mixed Traffic Scenarios

The DRL-based controllers heavily rely on traffic information obtained from CV technologies. However, achieving a pure CV environment may take a long time or may never be able to realize. It is more realistic to consider a mixed traffic environment where both CVs and NCVs travel on the road. Thus, it is crucial to evaluate the impact of CV's MPR on the performance of DRL-based controllers, as a controller that remains robust in mixed traffic environments would be more desirable. The statistical comparison of these DRL-based controllers, which are training in scenarios where the MPRs of CV are 100%, is depicted in Figure 4-10 and Figure 4-11.

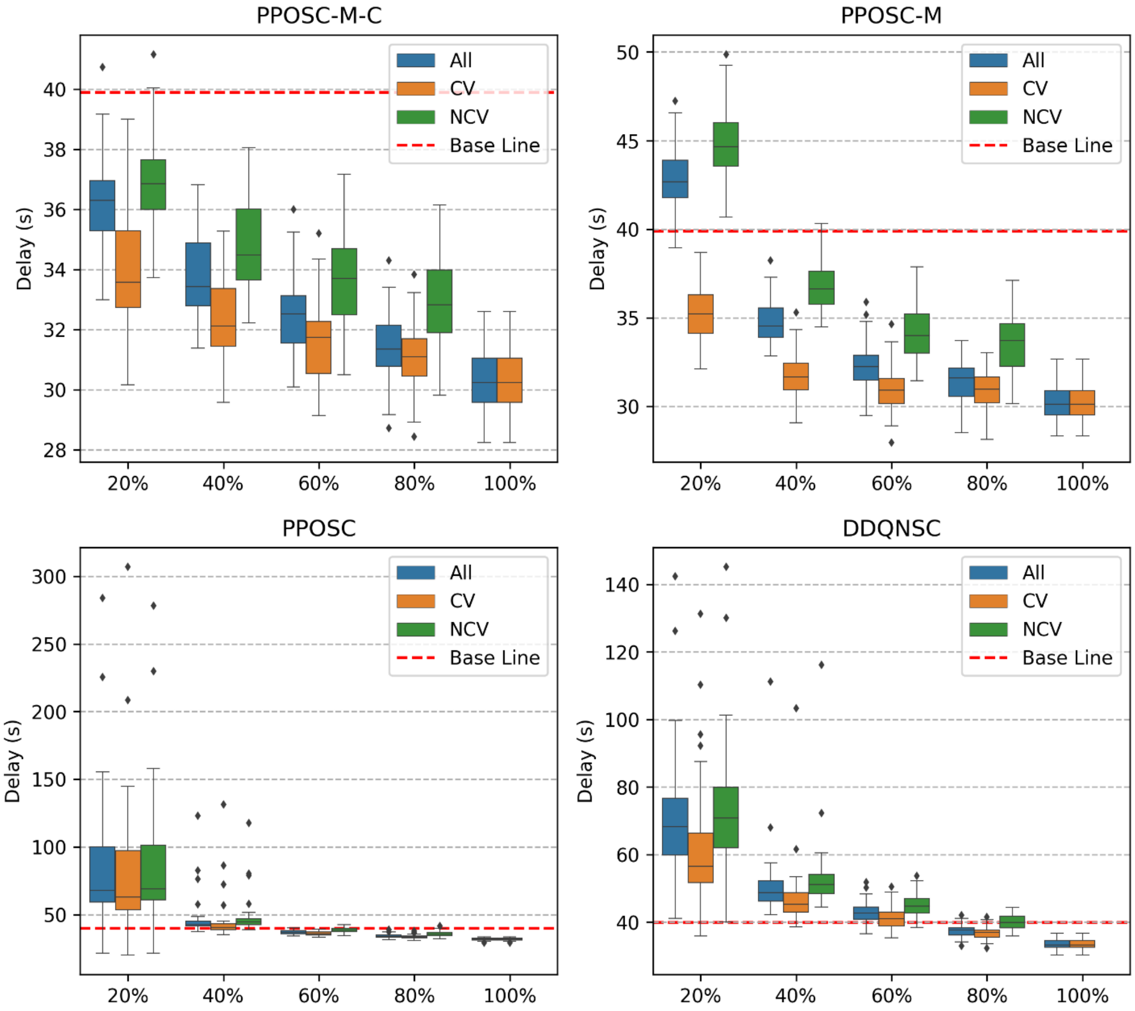


Figure 4-10 Delay Statistics under Different MPRs in Peak Scenarios at Isolated Intersection

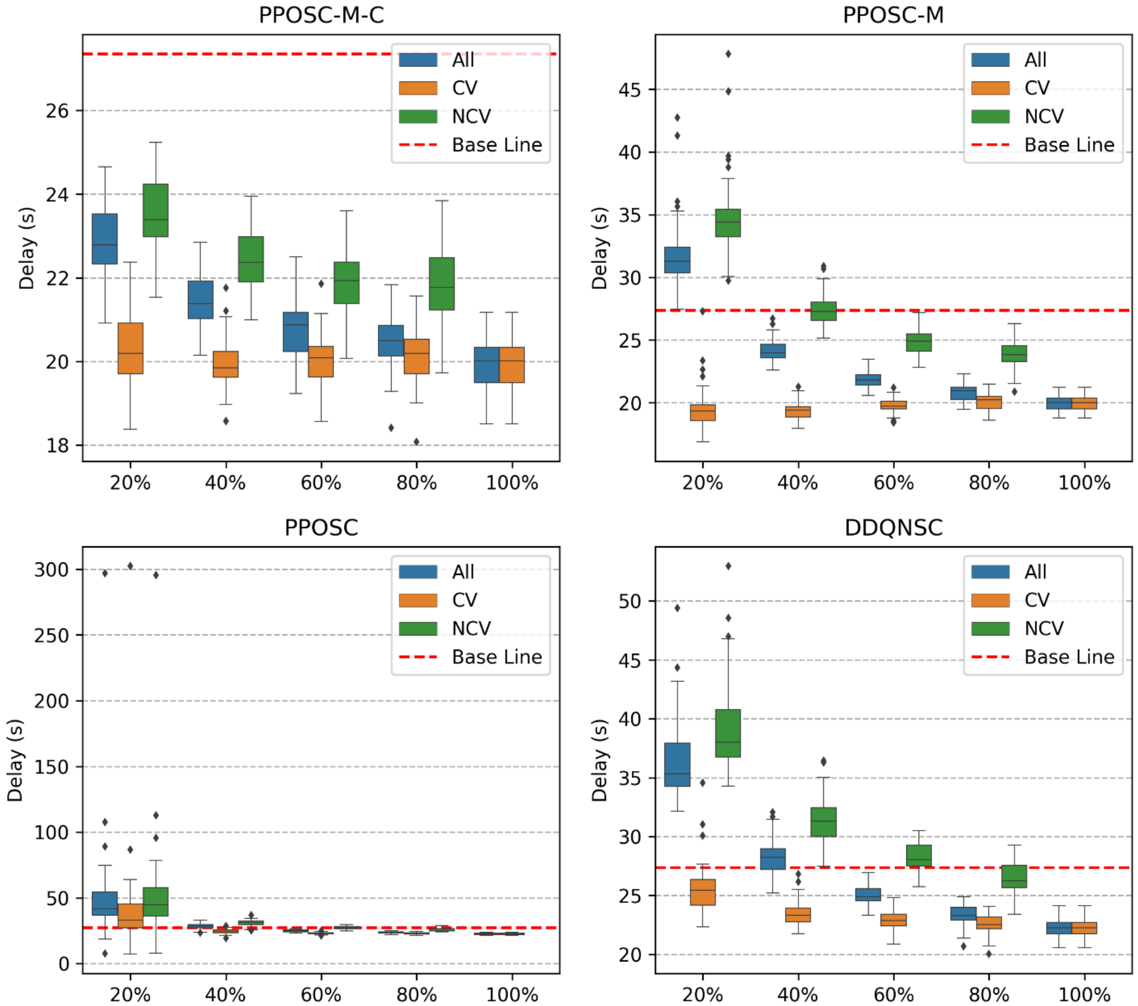


Figure 4-11 Delay Statistics under Different MPRs in Off-peak Scenarios at Isolated Intersection

In general, PPO-based controllers utilizing the multi-discrete action space have more robust performance regardless of the variation of MPR. This demonstrates that the implementation of the multi-discrete action space in PPO-based controllers is well suited for adoption in the field of TSC, especially in mixed traffic environments. This advantage may be due to the reduced frequency of decision-making, which leads to a significantly decreased sensitivity to the information obtained.

In addition, PPOSC-M-C outperforms PPOSC-M in terms of robustness. PPOSC-M-C can provide better services to NCVs while providing the same level of services to CVs compared to PPOSC-M, especially in low MPR scenarios. For example, in the peak demand with a 20% MPR scenario, the NCVs' average delay (36.92 s) is only 2.96 seconds greater than CVs' average delay (33.96 s) when controlled by PPOSC-M-C. In contrast, the average delay gap is

9.62 seconds between NCVs (44.86 s) and CVs (35.24 s) when controlled by PPOSC-M. This difference is even more obvious in the off-peak demand with a 20% MPR scenario, where the delay difference between NCVs and CVs is 3.22 seconds with PPOSC-M-C and 15.33 seconds with PPOSC-M. Besides, when using PPOSC-M-C, even in scenarios with as low as 20% MPR, the average delays of both CVs and NCVs are less than the baseline, represented by the average delays in scenarios controlled by the pretimed controller (39.90 s in the peak scenario and 27.35 s in the off-peak scenario). These results demonstrate that the implementation of the combined state space significantly enhances robustness in mixed traffic environments.

The comparison exhibits a notable deterioration in the performance of PPOSC in scenarios of 20% MPR, even compared to DDQNSC. However, in other scenarios, PPOSC and DDQNSC show similar levels of performance. This phenomenon suggests that PPO is more sensitive to the information it can observe compared to DDQN. Nevertheless, this issue can be effectively addressed by implementing the multi-discrete action space and the combined state space. This enhancement is likely due to the utilization of a broader set of actions and states, leading to improved adaptability across different scenarios, including those with sparse information availability. The results also indicate that, while all vehicles will experience reduced waiting times compared to traditional controllers, NCVs are expected to have longer average waiting times than CVs. This fact can be interpreted in a positive way, as suggested by Zhang et al. (2021), where this difference can incentivize people to equip their vehicles with connected functions and be more willing to share real-time vehicle information.

Chapter 5. Corridor

This chapter focuses on the experimental settings and results analysis in corridor scenarios.

5.1. Traffic Configuration

Two corridor scenarios are constructed to evaluate the performance of MARL-based signal controllers: a hypothetical scenario and a real-world scenario.

The hypothetical corridor consists of five identical intersections along the east-westbound direction, as shown in Figure 5-1. The distance between intersections is 500 meters. Each intersection is a four-leg intersection, with approaching roads having three lanes: one right-turn lane, one left-turn lane, and one through lane, as shown in Figure 5-2. The speed limit is 45 mph, which is 20 m/s. Buses travel in both the eastbound and westbound directions. Bus stops are positioned downstream of the intersections. The bus arrival headway is set at 5 minutes, which is implemented in the simulation by the time interval between buses entering the road network.

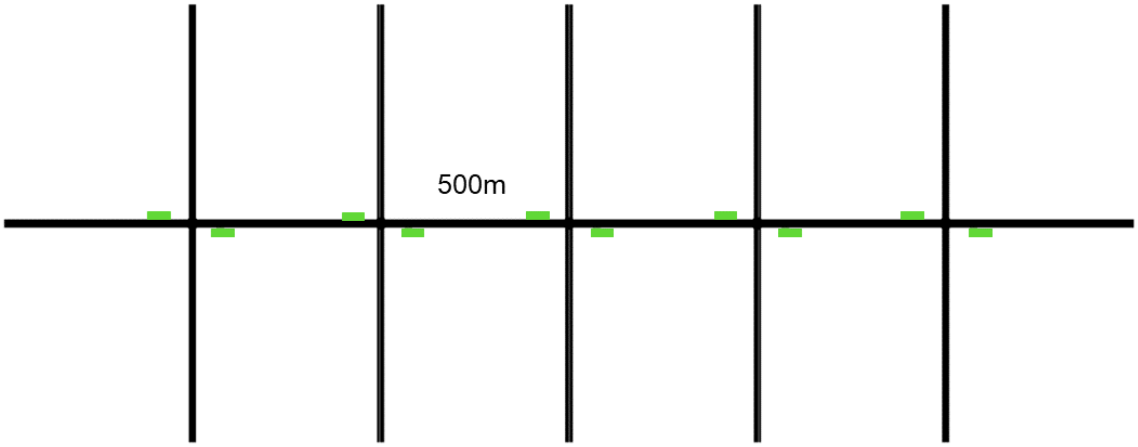


Figure 5-1 The Layout of the Hypothetical Corridor

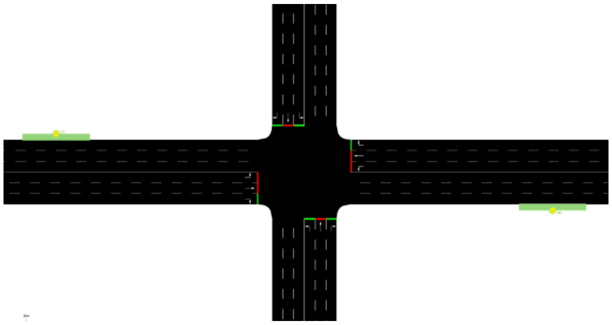


Figure 5-2 The Layout of the Hypothetical Intersection

The real-world corridor is modeled after the Central Avenue corridor located east of downtown Charlotte, North Carolina, USA. Five consecutive intersections have been selected and are listed from west to east: Central Avenue & Eastcrest Drive, Central Avenue & Briar Creek, Central Avenue & Eastway Drive, Central Avenue & Kilborne Drive, and Central Avenue & Rosehaven Drive. The corridor layout is illustrated in Figure 5-3. The distances between these intersections vary, ranging from 400 meters to 875 meters. The intersection layouts are further detailed in Figure 5-4. Among them, Intersection 1 is a T-shaped intersection, while the other intersections are four-leg intersections with varying numbers of approaching lanes and different channelization schemes. Otherwise, all other traffic settings remain consistent with those in the hypothetical corridor scenario.

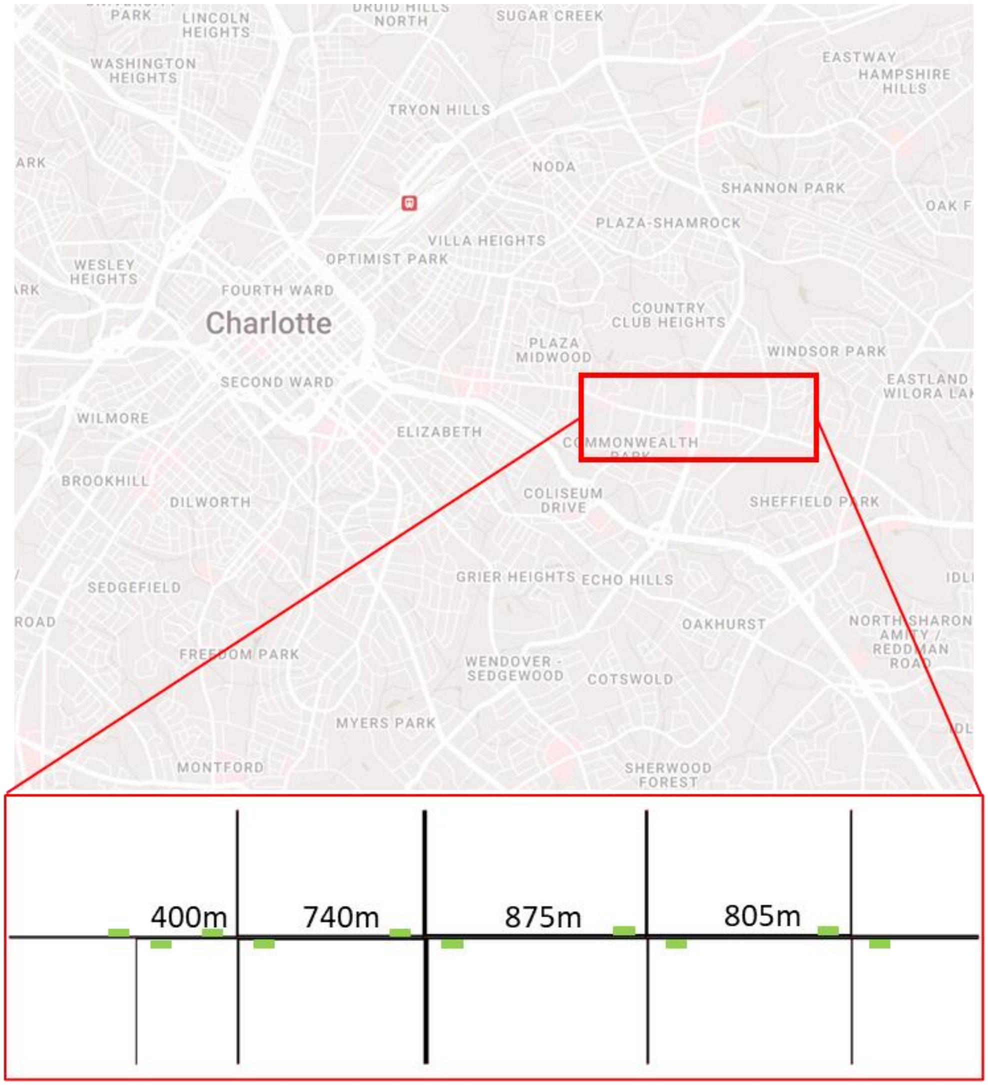


Figure 5-3 The Layout of the Real-World Corridor Scenario

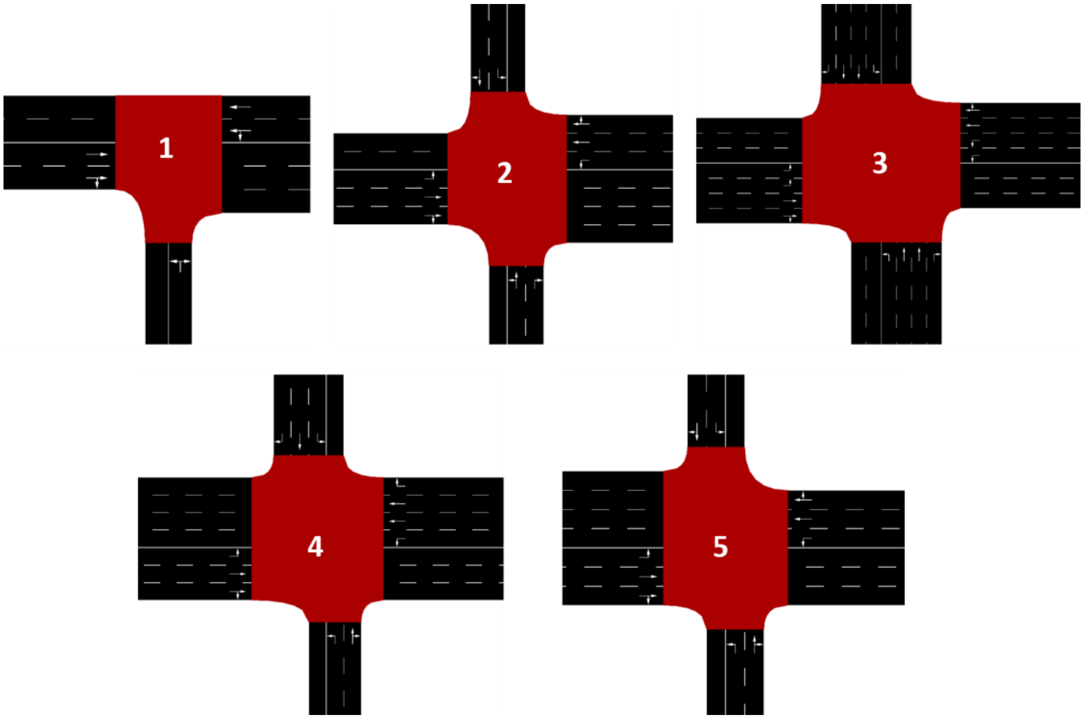


Figure 5-4 Layouts of the Intersections in the Real-World Corridor Scenario

5.2. Simulation Settings

We investigate two levels of traffic demand, namely high demand and low demand. In the hypothetical scenario, under high demand conditions, the traffic flow entering each boundary road is set at 850 veh/h, resulting in a total traffic volume of 10,200 veh/h along the corridor. In low demand conditions, 400 vehicles are inserted into each boundary road per hour, resulting in a total traffic demand of 4800 veh/h. Within the intersections, each approach road has the same turn ratio: a left-turning ratio of 10%, a through ratio of 75%, and a right-turning ratio of 15%.

For the real-world scenario, the peak hour volumes (representing high demand) and off-peak hour volumes (representing low demand) at the test corridor are utilized. The peak hour is from 5 to 6 PM on Wednesday, April 21, 2021, while the off-peak hour is from 9 to 10 AM on the same day. Traffic volume data is obtained from the Charlotte Department of Transportation (CDOT) and is presented Table 5-1. The traffic flows generated in the simulation follow a Poisson distribution. Each simulation run lasts one hour, with a ten-minute warm-up period.

Table 5-1 Traffic Volume Data in the Real-world Scenario, veh/h

Int.	Demand		SB			WB			NB			EB	
		R	T	L	R	T	L	R	T	L	R	T	L
1	PM Peak	-	-	-	-	583	3	62	-	29	50	984	-
	Off-peak	-	-	-	-	495	3	28	-	21	17	354	-
2	PM Peak	12	10	8	11	564	166	251	5	140	179	908	2

	Off-peak	3	6	6	1	421	158	68	9	92	107	324	2
3	PM Peak	176	793	88	68	341	206	325	883	180	193	547	246
	Off-peak	152	707	36	40	319	235	122	541	138	128	197	91
4	PM Peak	77	52	246	193	466	66	105	101	23	10	808	123
	Off-peak	79	42	128	116	486	41	24	39	8	12	298	51
5	PM Peak	41	10	50	23	676	101	38	27	66	69	1011	76
	Off-peak	44	9	27	21	560	73	23	16	41	22	416	26

Note: SB=southbound; WB=westbound; NB=northbound; EB=eastbound; R=right turn; T=Through, L=Left turn.

To comprehensively evaluate the effectiveness of the proposed MARL-based signal controllers, we have developed nine traffic signal controllers employing various signal control strategies. Specific settings for each controller are detailed below.

- **Pretimed Signal Controller (PSC):** The intersections are controlled by pretimed signal controllers utilizing the stage-based phase program, as shown in Figure 4-2(b). The signal timings are calculated using the Webster method (Koonce, 2008), based on the corresponding traffic volumes.

- **Actuated Signal Controller (ASC):** The intersections are controlled by fully actuated signal controllers, employing a typical National Electrical Manufacturers Association (NEMA) phase diagram. The phase sequence is shown in Figure 4-2(a).

- **Actuated Signal Controller with TSP Using CV (ATSP):** The intersections are controlled by fully actuated signal controllers, identical to ASC. However, a TSP strategy is activated when a bus approaches within 100 meters of the intersection. The TSP strategy implemented is an unconditional, active, rule-based strategy. Specifically, the traffic signal will switch to the appropriate phase upon activation to favor the buses progress. The buses are CVs and cars are HDVs.

The above three controllers represent conventional traffic signal controllers widely used in real-world practice. Their performance serves as baselines for real-world implementations.

- **Max Pressure Controller (MP):** This controller utilizes the Max Pressure strategy to control the SPaT at each intersection. Proposed by Varaiya (2013), this strategy introduces “pressure” as a metric to assess the traffic state at intersections. At each decision step, the phase with the maximum “pressure” will be chosen. Originally, “pressure” was defined as the product of link capacity and the difference in queue length between the input and the output links. Various formulations exist for pressure calculation. In this study, we employ the method proposed in Wei et al. (2019).

- **Max Pressure Controller with TSP (MP-TSP):** The SPaT at each intersection is controlled by the MP controller. However, this controller slightly modifies the pressure calculation by adding weights to buses. In this way, bus delays can be reduced.

- **Longest Queue First Controller (LQF):** Utilizing the Longest Queue First strategy introduced by Wunderlich et al. (2008), this controller regulates signals at each intersection with a straightforward control logic. At each decision step, the current queue length for each phase is obtained, and then the phase with the longest queue length is selected.

- **Longest Queue First Controller with TSP (LQF-TSP):** Similar to MP-TSP, this controller modifies the queue length calculation by adding weights to buses.

The above four controllers employ decentralized control strategies, and their decision-making processes have low computational cost, and therefore are easily scalable.

- **Multi-agent PPO (MAPPO):** This controller adopts the MARL framework to manage the SPaT at intersections. The control algorithm is PPO with necessary modifications to be compatible with multi-agent systems. Configurations pertaining to the global state, local observation, action space, and reward function are detailed in section 3.2.

- **Multi-agent PPO with Multi-discrete action (MAPPO-M):** MAPPO-M utilizes the multi-discrete action space. Aside from this modification, it shares the same configurations as MAPPO.

5.3. Training

Eight MARL controllers are trained, incorporating variations in controller configurations (MAPPO and MAPPO-M), corridor layouts (hypothetical corridor and real-world corridor), and traffic demands (high demand and low demand). Critical Python libraries used to implement these controllers include TracI, Gymnasium, PettingZoo, Pytorch, and RLlib. The Adam (adaptive moment estimation) optimizer is employed in the training process. The hyperparameters for both MAPPO and MAPPO-M have been well-tuned, and their values are presented in Table 5-2. All the MARL controllers are trained on a machine with Intel Core i7-11700 CPU, 32 GB of RAM, NVIDIA GeForce RTX 3080, and the Ubuntu 22.04.3 LTS operating system.

Table 5-2 Hyperparameters Used for MAPPO and MAPPO-M

Hyperparameter	Hypothetical Scenario	Real-world Scenario
Training episode	1000	1000
Discount factor	0.9	0.85
Learning rate	0.001 - 0.0003	0.001 - 0.0005
Train batch size	2048	2048
Batch size	256	256
Number of SGD iteration	3	3
Value function loss coefficient	0.5	0.5

Clip range	0.2	0.2
Entropy coefficient	0.01-0.001	0.01-0.001
Max gradient update	0.5	0.5

Their training performance for two types of corridors under both high and low traffic demand with 100% MPR is depicted in Figure 5-5 and Figure 5-6. MAPPO exhibits faster convergence and more stable performance compared to MAPPO-M, particularly under high-demand conditions. The utilization of the multi-discrete action space might make the training process more challenging in multi-agent systems, which differs from the phenomenon observed in single-agent systems in the previous chapter.

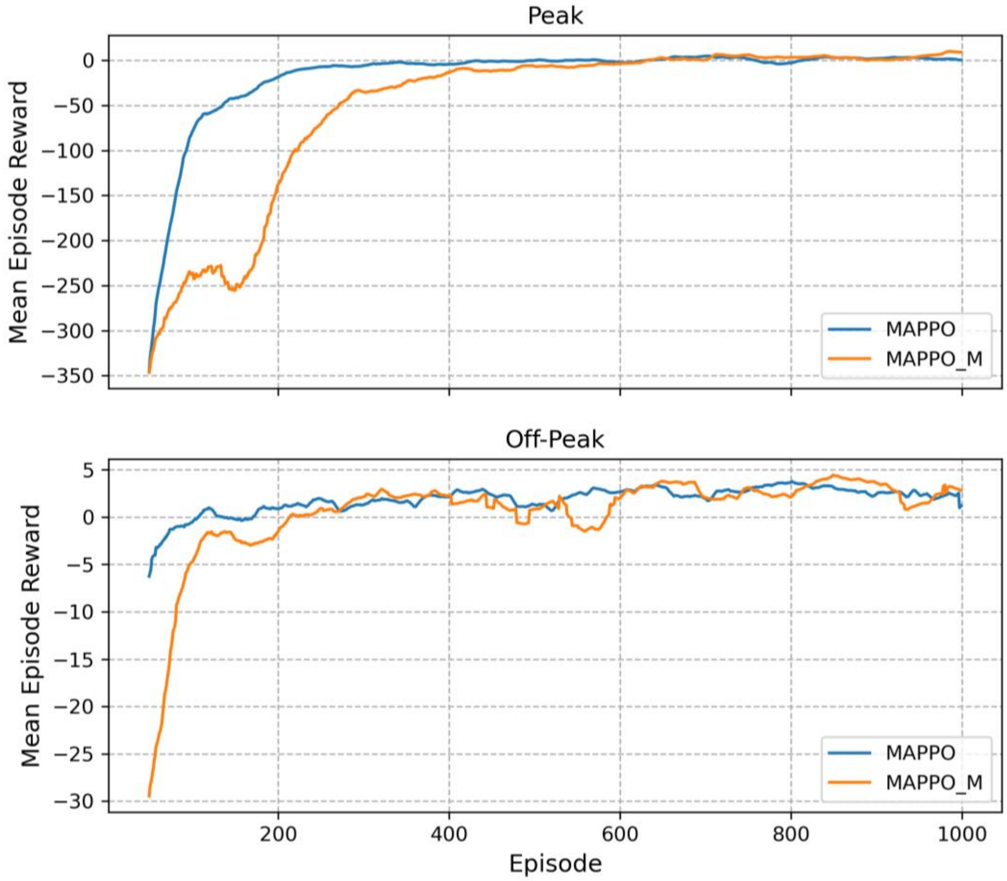


Figure 5-5 Mean Episode Reward Curves for MARL-based controllers under Peak and Off-peak Conditions in the Hypothetical Scenario

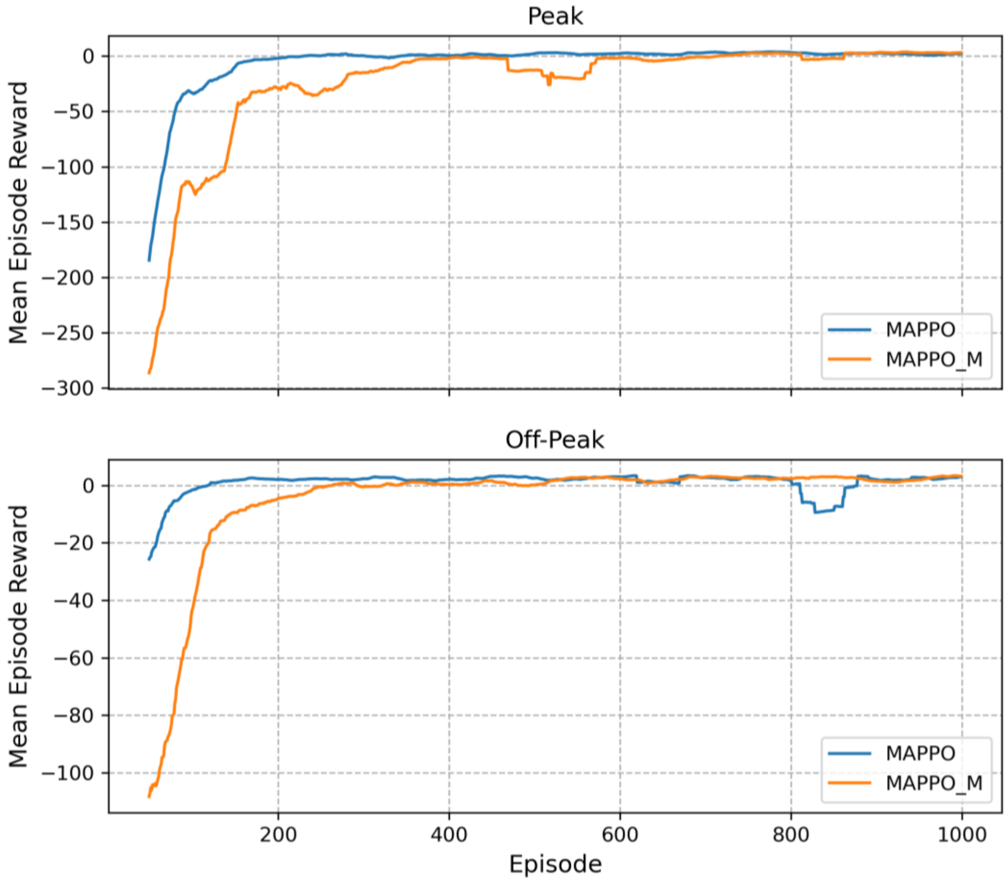


Figure 5-6 Mean Episode Reward Curves for MARL-based Controllers under Peak and Off-peak Conditions in the Real-world Scenario

5.4. Result Analyses

5.4.1. Performance Evaluation

We employ four metrics to comprehensively evaluate the performance of the controllers in corridor scenarios, with the performance of PSC serving as the baseline.

- **Average travel time (ATT):** The average travel time for all assessed targets from entering the road network to exiting the road network.

- **Average delay (AD)**: The average delays for all assessed targets travel in the road network.

- **Average speed (AS)**: The average speed for all assessed targets in the road network.

- **Average number of stops (ANS)**: The average number of stops for all assessed targets in the road network, excluding scheduled stops such as bus stops at the bus station.

A detailed performance comparison of the nine controllers in hypothetical corridor scenarios is presented in Table 5-3 and Table 5-4, while Table 5-5 and Table 5-6 display the results in real-world corridor scenarios.

5.4.1.1. Performance of Buses

First, the performance metrics of buses are examined. MAPPO-M demonstrates the best performance in terms of ATT, AD, AP, and ANS in almost all scenarios, regardless of testbeds and traffic demand conditions.

In the hypothetical corridor scenarios, when compared with the baseline, MAPPO-M reduces ATT, AD, and ANS by 52.17%, 78.26%, and 65.86%, respectively, under the high-demand condition. In low-demand conditions, it reduces ATT, AD, and ANS by 12.38%, 36.04%, and 49.75%, respectively. Additionally, the AS of buses increased by 108.77% and 13.26% under high and low demand conditions, respectively.

In the real-world corridor scenarios, compared to the baseline, MAPPO-M reduces ATT, AD, and ANS by 21.78%, 51.67%, and 62.18%, respectively, during peak hours. In addition, AS increased by 26.83%. During off-peak hours, compared with the baseline, ATT and AD decreased by 21.37% and 52.35%, respectively, while AP increased by 26.85%. When controlled by MAPPO-M, the ANS during off-peak hours is 1.57, slightly higher than ATSP but still better than the baseline with a reduction of 66.55%. Additionally, MAPPO also significantly improves the performance of buses in terms of mobility regardless of the testbeds and traffic demands, although its performance is slightly worse than MAPPO-M.

5.4.1.2. Performance of Cars

Regarding the performance of cars, fully actuated controllers exhibit superior performance.

In hypothetical scenarios, under high demand, ASC reduces ATT and AD by 13.29% and 24.19%, respectively. In low demand scenarios, ATSP performs the best, achieving reductions of 8.63% in ATT and 22.44% in AD. MP demonstrates the best performance in AS, with increases of 6.05% in high-demand and 5.89% in low-demand. MP also achieves the least ANS in the low-demand condition, which is 0.92, while PSC has the least ANS in the high demand condition, with a value of 1.04.

In real-world scenarios, fully actuated controllers, ASC and ATSP, continue to show the best performance in terms of ATT, AD, and AS. During peak hours, ASC reduces ATT and AD by 6.36% and 15.39%, respectively, and increases AS by 5.78% compared to the baseline. During off-peak hours, ATSP achieves reductions of 5.18% in ATT and 15.20% in AD, with an increase in AS by 4.78%. However, MAPPO-M outperforms other controllers in terms of ANS in both peak and off-peak conditions.

In terms of car performance, ASC performs the best in high-demand scenarios, while ATSP excels in low-demand scenarios. This is because in low-demand conditions, prioritizing buses has a less negative impact on other traffic. Additionally, MARL-based controllers, MAPPO and MAPPO-M, provide almost the same level of services to cars, which is desirable.

5.4.1.3. Performance of Person

In the hypothetical corridor scenarios under high-demand conditions, MAPPO-M outperforms other controllers in terms of ATT and AD, reducing ATT by 22.15% and AD by 38.92% compared to the baseline. It also ranks second in terms of AS and ANS among all controllers. Under low-demand conditions, ATSP has the best performance in terms of ATT, AD, and AS. Compared to the baseline, it reduces the ATT and AD by 7.72% and 20.89% respectively, while increasing AS by 5.28%. MAPPO-M has the least ANS, with a value of 0.98. MAPPO shows a similar level of capability to MAPPO-M, though with a slightly worse performance.

In real-world corridor scenarios, ATSP outperforms other controllers in terms of ATT, AD, and AS, while MAPPO-M has the best performance in terms of ANS. However, MARL-based controllers perform nearly as well as ATSP.

Overall, MAPPO-M and MAPPO demonstrate the best performance in terms of bus metrics, while ASC and ATSP excel in car metrics. Regarding metrics related to the average person, ATSP demonstrates superior performance, and MARL-based controllers also perform well. It is worth noting that MARL-based controllers can prioritize buses based on passenger occupancy, a capability lacking in conventional TSP controllers. Additionally, MARL-based controllers perform better in hypothetical scenarios than in real-world scenarios, possibly due to these problems having very different complexities. In hypothetical scenarios, traffic configurations remain identical across intersections, whereas in real-world scenarios, they vary significantly.

Table 5-3 Performance Comparison of Different Controllers in Hypothetical Corridor Scenarios under High Demand

Peak	Bus				Car				Person			
	ATT	AD	AS	ANS	ATT	AD	AS	ANS	ATT	AD	AS	ANS
PSC	749.43	499.56	4.00	5.58	151.12	83.91	10.85	1.04	195.39	114.66	10.34	1.38
Change rates	-	-	-	-	-	-	-	-	-	-	-	-
ASC	599.32	349.45	5.11	5.80	131.03	63.61	11.37	1.20	165.95	84.93	10.90	1.54
Change rates	-20.03%	-30.05%	27.77%	3.91%	-13.29%	-24.19%	4.77%	14.87%	-15.06%	-25.93%	5.39%	11.79%
ATSP	393.18	143.31	7.63	3.27	140.23	72.76	10.62	1.40	160.42	78.39	10.38	1.55
Change rates	-47.54%	-71.31%	90.67%	-41.43%	-7.21%	-13.29%	-2.08%	34.12%	-17.90%	-31.63%	0.40%	12.29%
MP	568.37	318.49	5.33	6.28	146.83	79.59	11.50	1.31	178.69	97.65	11.04	1.69
Change rates	-24.16%	-36.25%	33.12%	12.63%	-2.84%	-5.14%	6.05%	25.95%	-8.55%	-14.84%	6.73%	22.53%
MP-TSP	402.52	152.64	7.47	3.45	151.03	83.75	10.75	1.34	171.19	89.28	10.49	1.51
Change rates	-46.29%	-69.44%	86.80%	-38.22%	-0.06%	-0.18%	-0.90%	28.58%	-12.38%	-22.14%	1.41%	9.52%
LQF	553.91	304.03	5.49	6.02	150.98	83.72	10.47	1.51	181.41	100.35	10.09	1.85
Change rates	-26.09%	-39.14%	37.20%	7.98%	-0.10%	-0.22%	-3.47%	45.03%	-7.15%	-12.47%	-2.38%	34.44%
LQF-TSP	500.51	250.64	6.06	5.61	152.86	85.59	10.26	1.51	179.67	98.31	9.93	1.83
Change rates	-33.21%	-49.83%	51.43%	0.59%	1.15%	2.01%	-5.45%	45.02%	-8.04%	-14.25%	-3.95%	32.63%
MAPPO	359.20	109.31	8.34	1.98	138.11	70.55	10.54	1.46	155.74	73.64	10.37	1.50
Change rates	-52.07%	-78.12%	108.33%	-64.44%	-8.61%	-15.92%	-2.82%	39.62%	-20.29%	-35.77%	0.24%	8.69%
MAPPO-M	358.46	108.59	8.35	1.90	134.21	66.69	10.97	1.34	152.10	70.04	10.76	1.39
Change rates	-52.17%	-78.26%	108.77%	-65.86%	-11.19%	-20.51%	1.10%	28.69%	-22.15%	-38.92%	4.04%	0.62%

Note: ATT represents average travel time (s); AD represents average delay (s); AS represents average speed (m/s); ANS represents average number of stops.

Table 5-4 Performance Comparison of Different Controllers in Hypothetical Corridor Scenarios under Low Demand

Off-peak	Bus				Car				Person			
	ATT	AD	AS	ANS	ATT	AD	AS	ANS	ATT	AD	AS	ANS
PSC	380.52	130.66	7.92	2.02	110.92	43.24	13.26	1.18	152.76	56.81	12.43	1.31
Change rates	-	-	-	-	-	-	-	-	-	-	-	-
ASC	419.93	170.04	7.17	4.14	101.43	33.62	13.98	1.02	150.72	54.74	12.92	1.50
Change rates	10.36%	30.14%	-9.47%	105.36%	-8.56%	-22.24%	5.38%	-13.99%	-1.33%	-3.64%	3.94%	14.39%
ATSP	356.99	107.12	8.38	1.81	101.35	33.54	13.95	1.04	140.97	44.94	13.09	1.16
Change rates	-6.18%	-18.02%	5.85%	-10.45%	-8.63%	-22.44%	5.20%	-12.16%	-7.72%	-20.89%	5.28%	-11.76%
MP	420.62	170.73	7.16	3.96	107.59	39.83	14.04	0.92	156.03	60.09	12.98	1.39
Change rates	10.54%	30.67%	-9.56%	96.14%	-3.00%	-7.88%	5.89%	-22.46%	2.14%	5.78%	4.39%	5.73%
MP-TSP	346.29	96.41	8.64	1.82	108.21	40.44	13.85	0.93	145.16	49.13	13.04	1.06
Change rates	-9.00%	-26.21%	9.16%	-9.67%	-2.44%	-6.46%	4.44%	-21.77%	-4.98%	-13.51%	4.91%	-18.88%
LQF	426.94	177.07	7.05	4.11	109.84	42.08	13.20	1.17	158.96	62.99	12.25	1.62
Change rates	12.20%	35.52%	-10.92%	103.75%	-0.97%	-2.68%	-0.45%	-1.15%	4.06%	10.88%	-1.47%	23.81%
LQF-TSP	409.14	159.27	7.34	4.09	110.19	42.43	13.15	1.17	156.56	60.55	12.25	1.62
Change rates	7.52%	21.90%	-7.33%	102.73%	-0.66%	-1.87%	-0.82%	-1.27%	2.49%	6.60%	-1.46%	23.52%
MAPPO	345.54	95.68	8.66	1.62	108.14	40.37	13.48	0.94	145.00	48.95	12.74	1.05
Change rates	-9.19%	-26.77%	9.42%	-19.79%	-2.50%	-6.64%	1.67%	-20.37%	-5.08%	-13.82%	2.44%	-20.22%
MAPPO-M	333.68	83.81	8.96	1.08	107.86	40.05	13.41	0.96	142.87	46.84	12.72	0.98
Change rates	-12.31%	-35.85%	13.19%	-46.53%	-2.76%	-7.36%	1.14%	-18.61%	-6.47%	-17.55%	2.33%	-25.27%

Note: ATT represents average travel time (s); AD represents average delay (s); AS represents average speed (m/s); ANS represents average number of stops.

Table 5-5 Performance Comparison of Different Controllers in Real-world Corridor Scenarios under High Demand

Peak	Bus				Car				Person			
	ATT	AD	AS	ANS	ATT	AD	AS	ANS	ATT	AD	AS	ANS
PSC	502.94	212.04	7.65	4.22	164.44	69.08	11.97	1.60	203.65	85.64	11.47	1.90
Change rates	-	-	-	-	-	-	-	-	-	-	-	-
ASC	485.92	195.02	7.90	4.05	153.98	58.45	12.66	1.50	192.29	74.20	12.11	1.79
Change rates	-3.39%	-8.03%	3.26%	-4.13%	-6.36%	-15.39%	5.78%	-6.20%	-5.58%	-13.36%	5.60%	-5.74%
ATSP	408.93	118.01	9.33	2.09	154.46	58.91	12.57	1.53	184.65	65.92	12.19	1.60
Change rates	-18.69%	-44.34%	21.98%	-50.55%	-6.07%	-14.72%	5.01%	-4.32%	-9.33%	-23.03%	6.24%	-16.12%
MP	480.57	189.68	7.99	4.19	178.76	83.16	11.59	1.62	213.96	95.58	11.17	1.92
Change rates	-4.45%	-10.55%	4.36%	-0.82%	8.71%	20.39%	-3.18%	1.02%	5.06%	11.61%	-2.62%	0.64%
MP-TSP	417.96	127.06	9.15	2.38	199.84	104.26	10.57	1.65	225.90	106.98	10.40	1.74
Change rates	-16.90%	-40.08%	19.58%	-43.58%	21.53%	50.93%	-11.68%	3.03%	10.92%	24.92%	-9.30%	-8.81%
LQF	496.56	205.66	7.73	4.63	170.37	74.97	11.57	1.92	207.99	90.04	11.13	2.24
Change rates	-1.27%	-3.01%	1.05%	9.70%	3.61%	8.53%	-3.37%	20.22%	2.13%	5.14%	-3.01%	17.43%
LQF-TSP	474.56	183.65	8.07	4.61	177.28	81.86	11.08	1.97	212.09	93.78	10.73	2.28
Change rates	-5.64%	-13.39%	5.42%	9.26%	7.81%	18.50%	-7.44%	23.19%	4.14%	9.50%	-6.48%	19.78%
MAPPO	405.48	114.58	9.42	2.17	155.69	60.00	12.49	1.48	185.31	66.48	12.12	1.56
Change rates	-19.38%	-45.96%	23.07%	-48.67%	-5.32%	-13.14%	4.28%	-7.75%	-9.01%	-22.38%	5.66%	-18.17%
MAPPO-M	393.38	102.48	9.70	1.60	158.59	62.86	12.31	1.45	186.44	67.56	12.00	1.47
Change rates	-21.78%	-51.67%	26.83%	-62.18%	-3.56%	-9.00%	2.81%	-9.49%	-8.45%	-21.11%	4.60%	-23.00%

Note: ATT represents average travel time (s); AD represents average delay (s); AS represents average speed (m/s); ANS represents average number of stops.

Table 5-6 Performance Comparison of Different Controllers in Real-world Corridor Scenarios under Low Demand

Off-peak	Bus				Car				Person			
	ATT	AD	AS	ANS	ATT	AD	AS	ANS	ATT	AD	AS	ANS
PSC	491.49	200.58	7.78	4.68	140.28	48.59	13.27	1.45	196.62	72.98	12.39	1.97
Change rates	-	-	-	-	-	-	-	-	-	-	-	-
ASC	458.74	167.84	8.36	3.78	133.16	41.37	13.94	1.29	188.71	62.95	12.98	1.71
Change rates	-6.66%	-16.32%	7.43%	-19.18%	-5.07%	-14.86%	5.03%	-11.04%	-4.02%	-13.74%	4.81%	-12.85%
ATSP	395.62	104.71	9.64	1.54	133.00	41.21	13.90	1.30	178.19	52.13	13.17	1.34
Change rates	-19.51%	-47.80%	23.84%	-67.21%	-5.18%	-15.20%	4.78%	-10.04%	-9.37%	-28.56%	6.30%	-31.73%
MP	448.13	157.22	8.56	3.49	145.30	53.50	13.31	1.25	197.10	71.24	12.50	1.63
Change rates	-8.82%	-21.62%	10.05%	-25.49%	3.58%	10.09%	0.30%	-13.96%	0.24%	-2.38%	0.88%	-17.14%
MP-TSP	400.19	109.30	9.54	2.10	163.63	71.86	11.94	1.44	204.45	78.32	11.53	1.55
Change rates	-18.58%	-45.51%	22.60%	-55.24%	16.65%	47.88%	-10.00%	-0.58%	3.98%	7.33%	-6.95%	-21.04%
LQF	465.31	174.39	8.23	4.17	142.39	50.68	13.09	1.53	197.23	71.69	12.26	1.98
Change rates	-5.33%	-13.06%	5.74%	-10.90%	1.51%	4.28%	-1.35%	5.65%	0.31%	-1.77%	-1.00%	0.59%
LQF-TSP	456.26	165.35	8.38	4.36	149.55	57.86	12.45	1.60	202.19	76.31	11.75	2.07
Change rates	-7.17%	-17.56%	7.65%	-6.92%	6.61%	19.07%	-6.17%	10.22%	2.83%	4.57%	-5.14%	5.26%
MAPPO	391.11	100.21	9.75	1.83	139.00	47.18	13.35	1.27	182.39	56.30	12.73	1.36
Change rates	-20.42%	-50.04%	25.35%	-60.99%	-0.91%	-2.92%	0.62%	-12.51%	-7.24%	-22.85%	2.78%	-30.68%
MAPPO-M	386.48	95.58	9.87	1.57	139.84	48.03	13.24	1.23	182.30	56.22	12.66	1.29
Change rates	-21.37%	-52.35%	26.85%	-66.55%	-0.31%	-1.15%	-0.24%	-14.87%	-7.28%	-22.96%	2.17%	-34.39%

Note: ATT represents average travel time (s); AD represents average delay (s); AS represents average speed (m/s); ANS represents average number of stops.

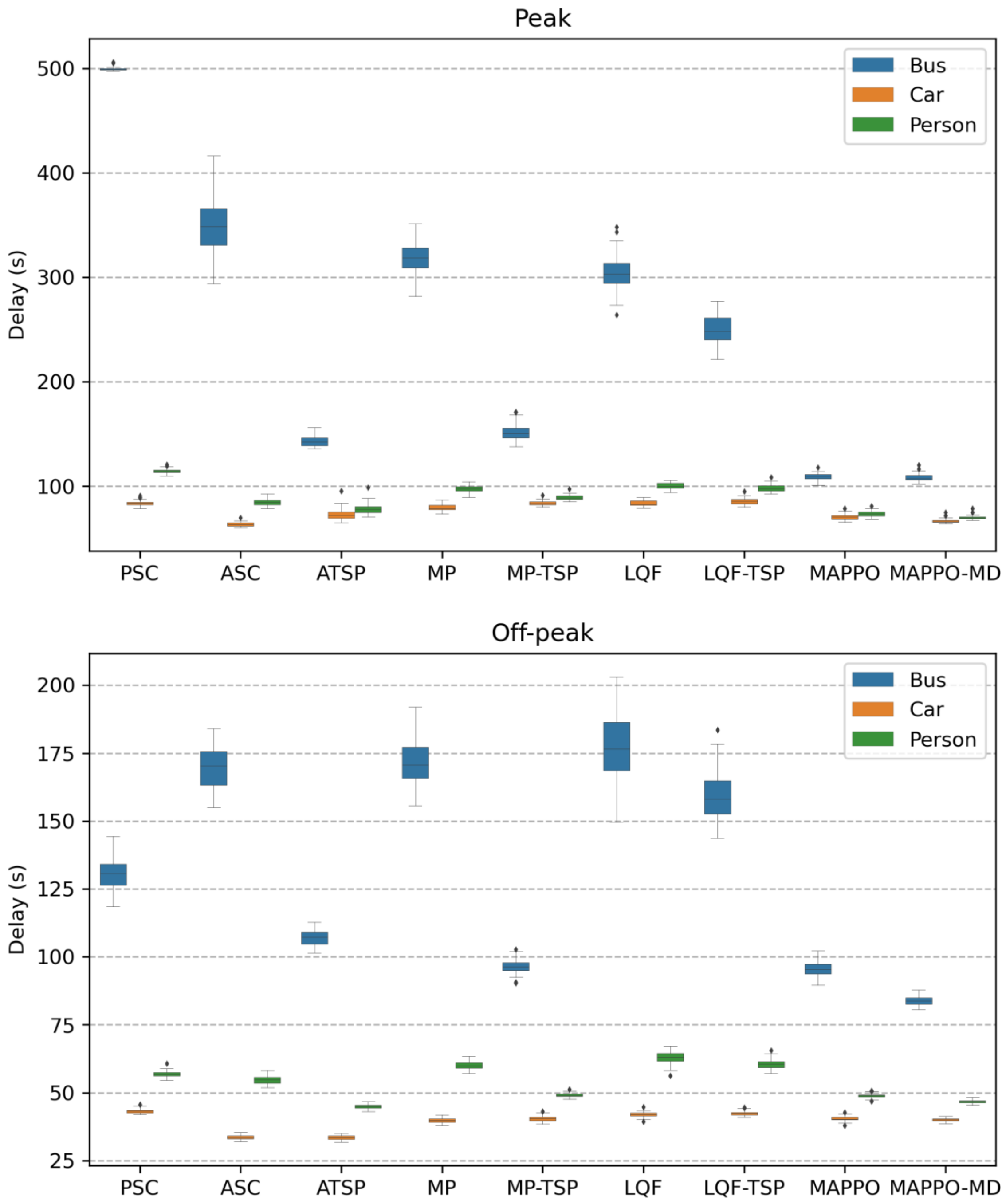


Figure 5-7 Average Delay Statistics in Different Controllers in Hypothetical Corridor Scenarios

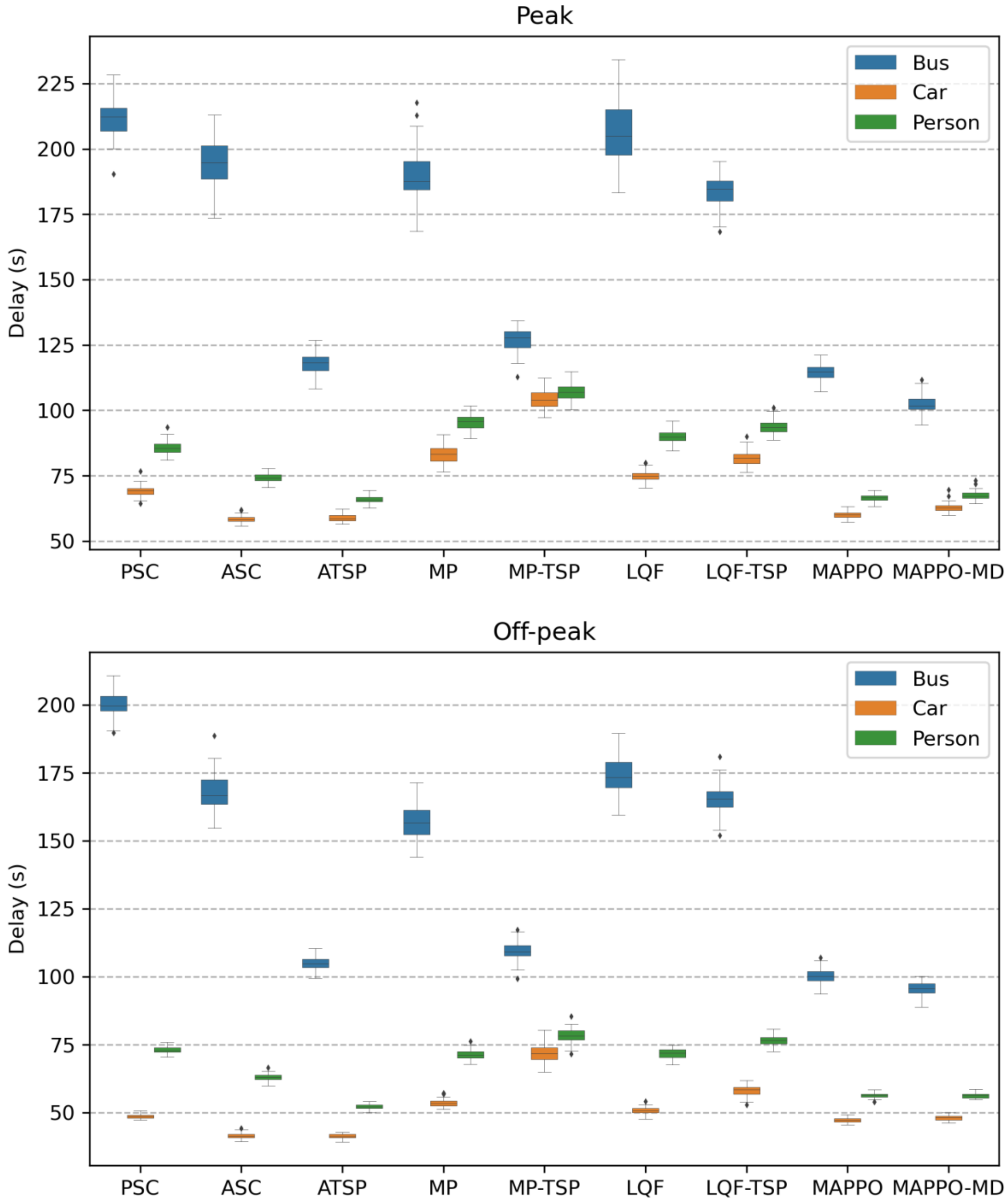


Figure 5-8 Average Delay Statistics in Different Controllers in Real-world Corridor Scenarios

Figure 5-7 and Figure 5-8 illustrate average delay statistics for different controllers in different scenarios. Generally, controllers without TSP exhibit greater variances in average bus delay compared to controllers with TSP. ATPS, MP-TSP, MAPPO, and MAPPO-M significantly reduce the average bus delays, with MAPPO-M performs the best and demonstrating the smallest variance. In real-world scenarios, MP-TSP and LQF-TSP greatly increase average car delay. However, in hypothetical scenarios, their performance in terms of average car delay is not significantly compromised. This may suggest that these two controllers struggle to balance bus priority with car services under unbalanced traffic demand conditions. These figures further emphasize that MARL-based controllers provide stable services for buses.

5.4.2. Sensitivity Analysis

5.4.2.1. CV Market Penetration Rate

In corridor scenarios, we also examine the impact of the CV market penetration rate on the performance of the proposed controllers. Ten scenarios, encompassing both peak and off-peak hours, have been designed with the MPR ranging from 20% to 100%, in increments of 20%. Other settings remain consistent with the basic scenarios. The results are shown in Figure 5-9 and Figure 5-10.

As the MRP increases, both MAPPO-M and MAPPO demonstrate improved performance in terms of average delays across all scenarios. This improvement becomes more pronounced when the MPR is lower than 60%. In hypothetical scenarios, the average person delay is lower than the baseline when the MRP reaches 60%, while in real-world scenarios, the average person delay is lower than the baseline when the MPR reaches 40%. The MAPPO controller trained in the hypothetical scenario with high traffic demand outperforms MAPPO-M in low MPR scenarios. However, MAPPO-M outperforms MAPPO in the other three scenarios when the MPR is low, providing more stable services for both buses and cars. This suggests that the introduction of multi-discrete action space can enhance the robustness of MARL-based controllers in mixed traffic environments, aligning with our findings in isolated intersection scenarios.

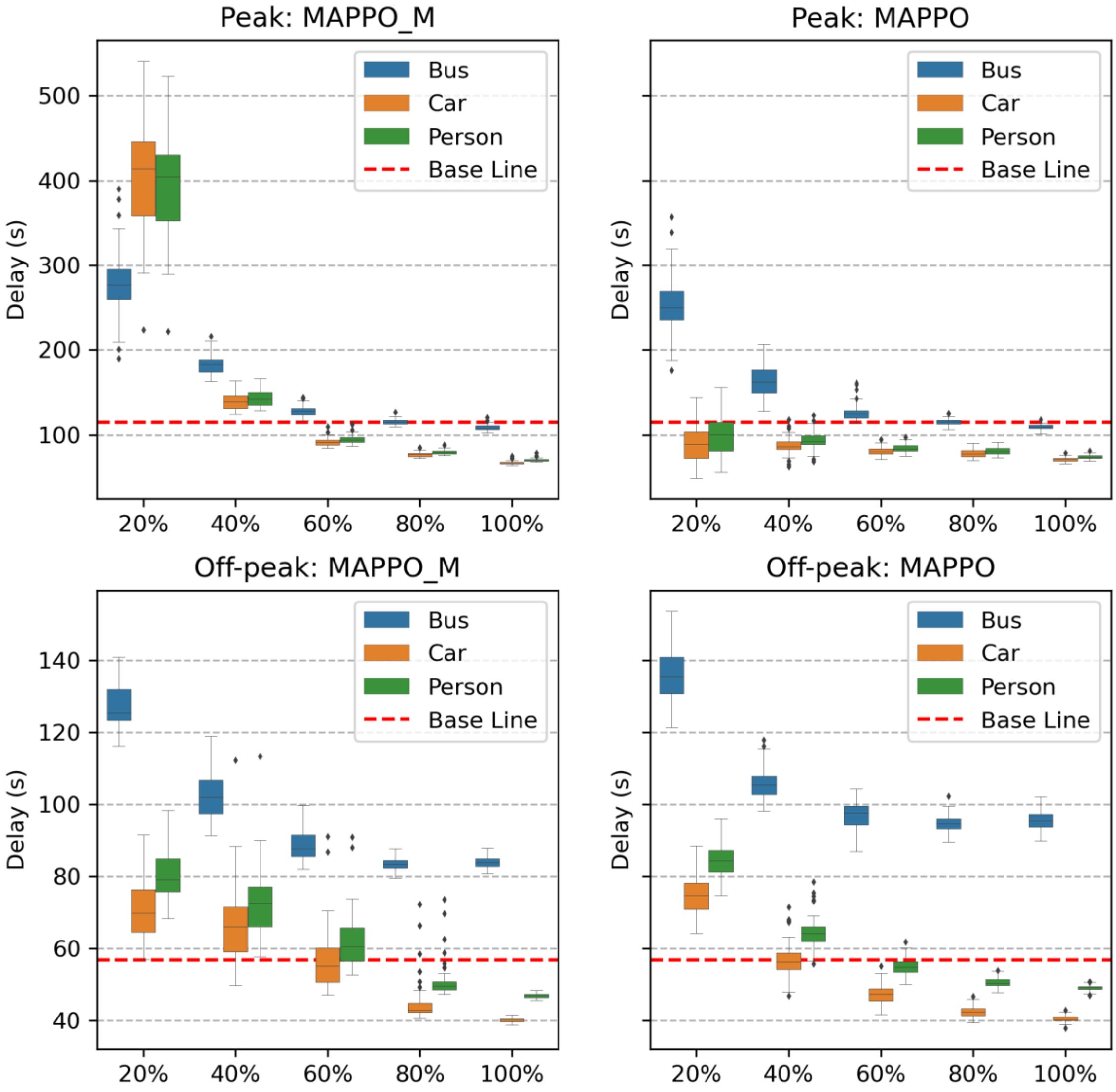


Figure 5-9 Sensitivity of Controllers to CV Market Penetration Rate in Hypothetical Corridor Scenarios

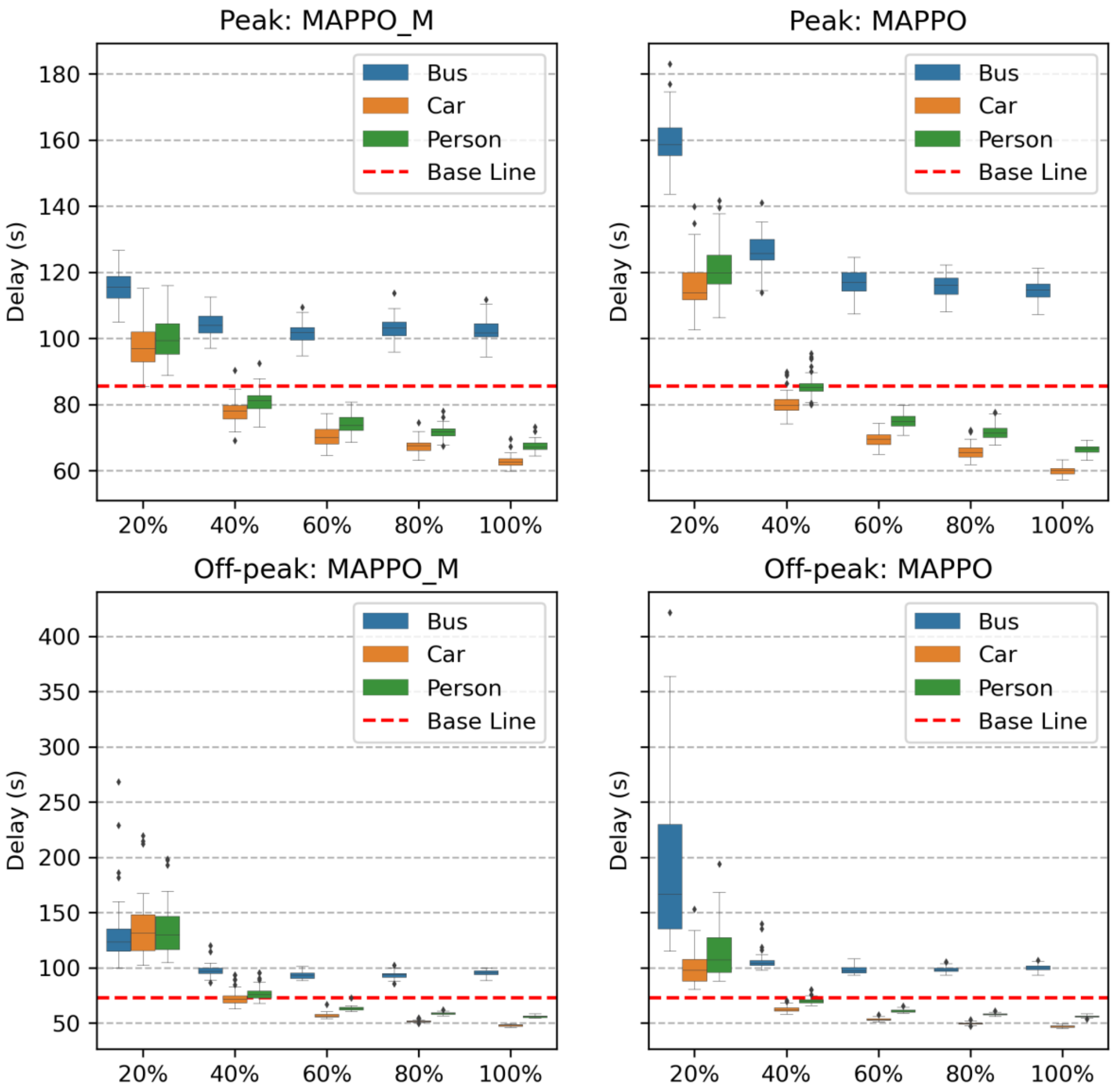


Figure 5-10 Sensitivity of Controllers to CV Market Penetration Rate in Real-world Corridor Scenarios

5.4.2.2. Bus Passenger Occupancy

In this section, we investigate the sensitivity to bus passenger occupancy by varying the number of passengers per bus across scenarios. All other settings remain the same as in the basic scenarios. Specifically, we set the number of passengers on each bus to 1, 10, 30, 50, and 70 passengers.

The results, as shown in Figure 5-11 and Figure 5-12, indicate that as bus passenger occupancy increases from 1 passenger per bus to 30 passengers per bus, the average bus delay decreases significantly across all scenarios. When the passengers on the bus exceed 30, the MARL-based controllers become insensitive to it, especially during peak hours. The average car delay experiences a slight increase with more priority given to buses. However, these increases are minimal compared to the reduction in average bus delays, which suggests that the adverse impacts on the car are likely to be negligible, even with buses receiving more priority. Compared to MAPPO, except for the hypothetical scenario during peak hours with bus passenger occupancy less than 30, MAPPO-M provides better service for buses.

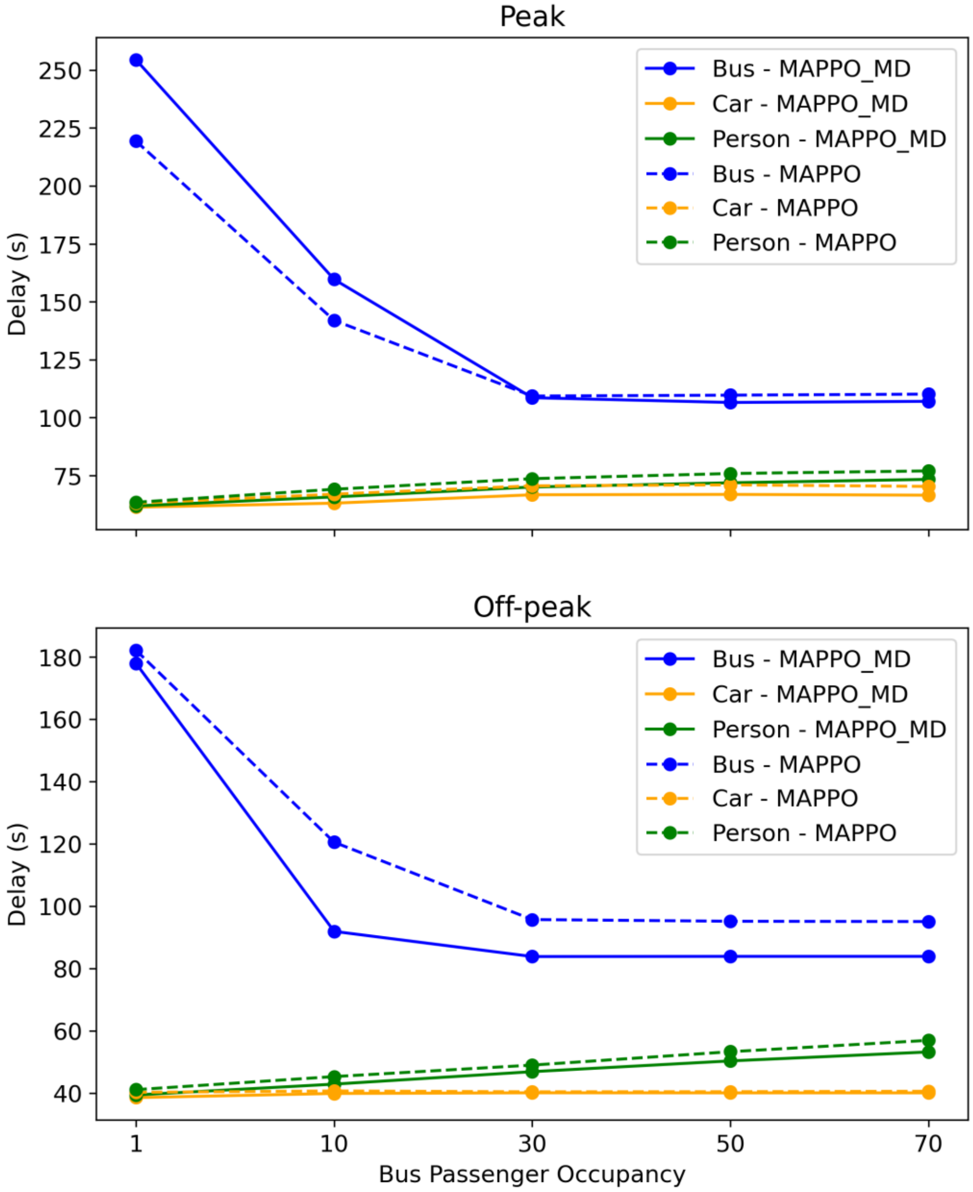


Figure 5-11 Sensitivity of Controllers to Bus Passenger Occupancy in Hypothetical Corridor Scenarios

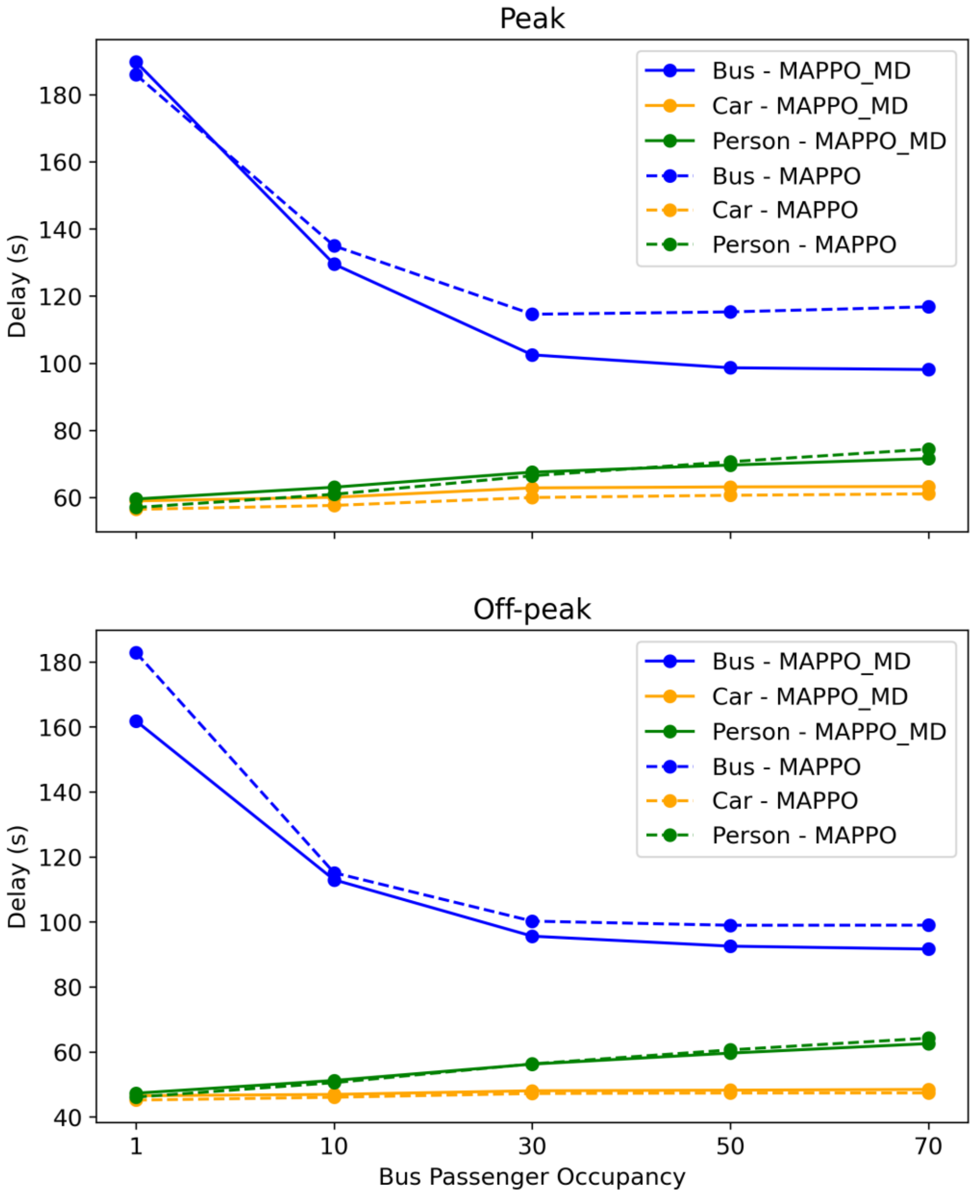


Figure 5-12 Sensitivity of Controllers to Bus Passenger Occupancy in Real-world Corridor Scenario

5.4.2.3. Bus Arrival Headway

This section explores the impact of bus arrival headways on the effectiveness of the proposed controllers by considering five different headways: 2 minutes, 5 minutes, 10 minutes, 15 minutes, and 30 minutes. The rest of the scenario settings align with the basic scenarios.

As illustrated in Figure 5-13 and Figure 5-14, changes in bus arrival headway have insignificant impacts on the average bus delay, especially in scenarios controlled by MAPPO-M. As bus arrivals become less frequent, the average car delay decreases, as well as the average person delay. This occurs because a reduced frequency of bus arrivals results in less interruption to traffic, as fewer buses requiring priority.

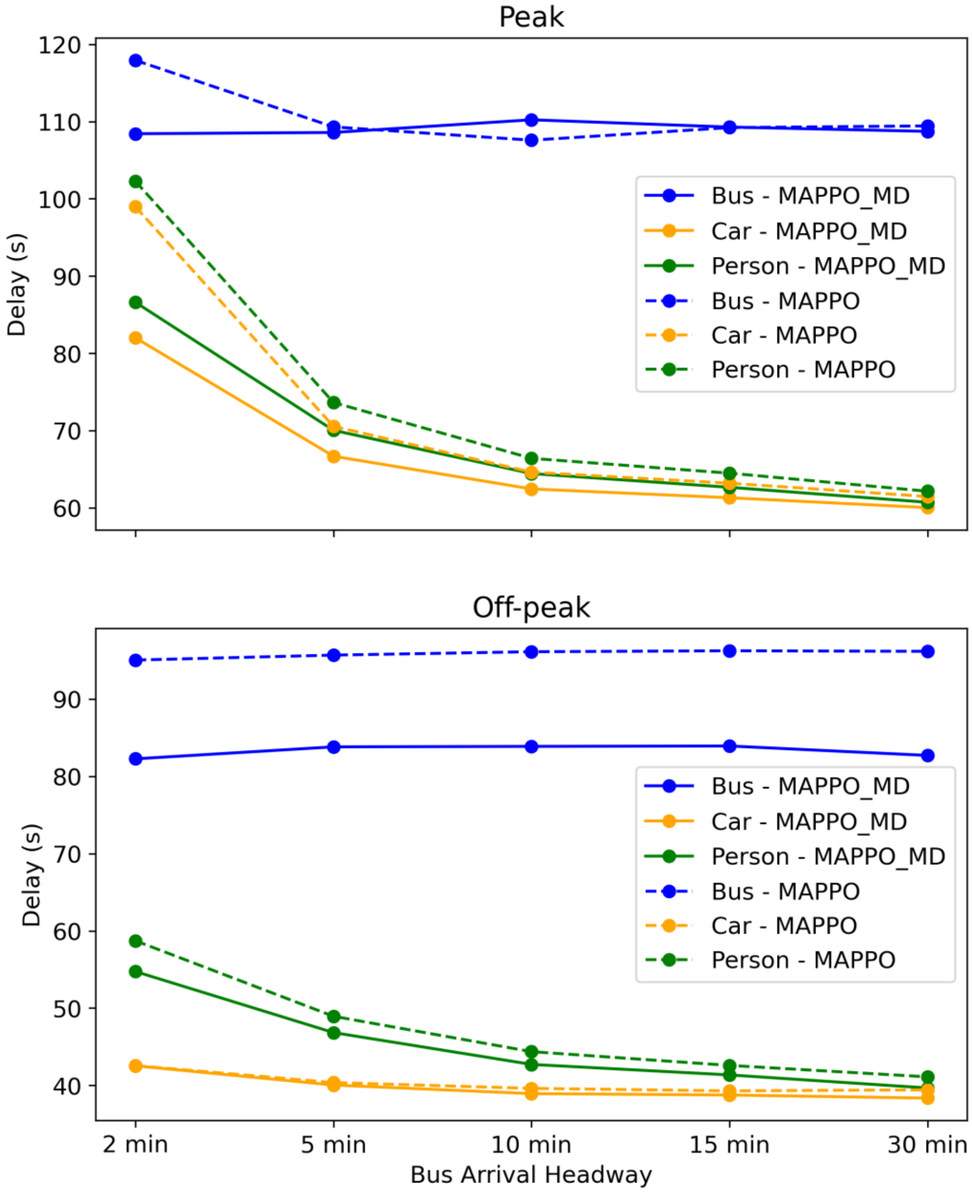


Figure 5-13 Sensitivity of Controllers to Bus Arrival Headway in Hypothetical Corridor Scenarios

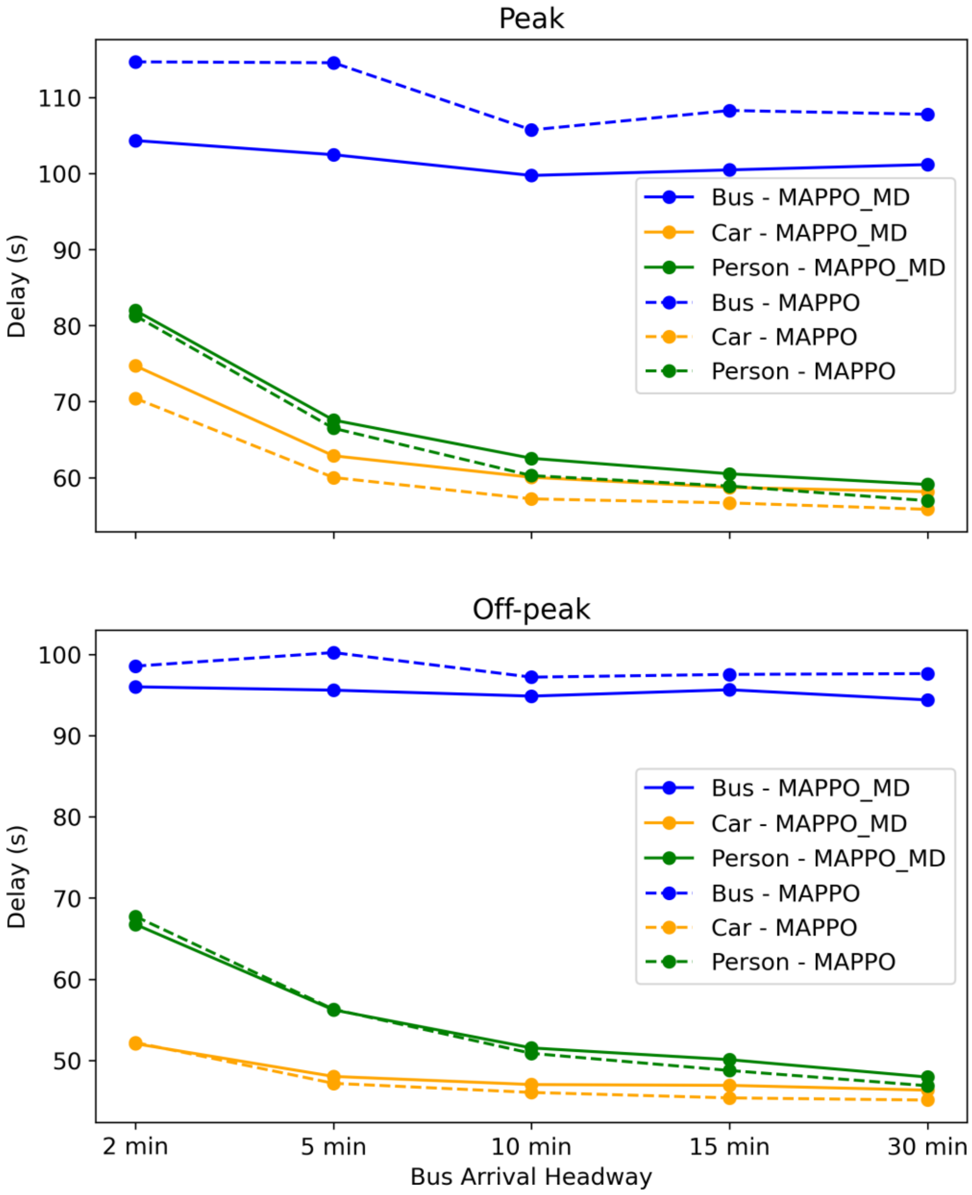


Figure 5-14 Sensitivity of Controllers to Bus Arrival Headway in Real-world Corridor Scenario

Chapter 6. Conclusions

This study begins with a comprehensive literature review about optimization algorithms in TSC and TSP, which reveals research gaps and provides the basis for our contribution. The following chapter outlines the methodology adopted in this study, including DRL algorithms and MARL algorithms. Detailed configurations of RL algorithms are also covered, including global state spaces, local observation spaces, action spaces, reward functions, and neural network structures. The core of our work is the development of adaptive TSP controllers utilizing DRL algorithms and leveraging real-time data obtained through CV technology. These controllers, proposed in this study, aim to overcome the limitations of existing conventional controllers as well as state-of-the-art controllers. The study can be divided into two parts: (1) isolated intersection-level DRL-based TSP control and (2) corridor-level MARL-based TSP control. At the isolated intersection level, we introduce the DQN-TSP controller, designed to prioritize transit vehicles while mitigating adverse effects on the whole traffic system. We also delve into the aspect of enhancing robustness in mixed traffic environments. This is achieved by leveraging multiple data sources and introducing innovative action spaces. Therefore, we proposed the PPOSC-M-C controller, which is built on PPO and utilizes traffic information from both CVs and cameras within the intersection. Besides, this controller adopts a novel action space, multi-discrete action space, which is seldom used in the TSC research area, aiming to obtain better performance. At the corridor level, we introduced MARL-based controllers to manage individual intersections while ensuring coordination between them. Simultaneously, the proposed MARL-based controllers prioritize bus progress. These controllers utilize the PPO algorithm with minor modifications. Specifically, it is adapted to be compatible with the framework of centralized training and decentralized execution, enhancing overall performance. Additionally, parameter sharing, which is an efficient training technique, is also implemented to achieve faster and more stable training processes. To comprehensively evaluate the proposed controllers, we built three simulation testbeds, i.e., the real-world isolated intersection testbed, the real-world corridor testbed, and the hypothetical corridor testbed. Subsequently, numerous experiments were conducted within the testbeds to explore various scenarios involving different traffic demands, signal control strategies, etc. Additionally, we delved into sensitivity analysis, focusing on critical factors such as CV market penetration rates, bus passenger occupancies, and bus arrival headways.

For isolated intersection scenarios, results indicate that the proposed DQN controller outperforms the conventional controllers in terms of average person delay. Compared to the baseline, it reduces the average person delay by 18.77% and 23.37% in both peak and off-peak conditions, respectively. The proposed controller also leads to a decrease in both average bus and car delays. Although the fully actuated controller with TSP performs better in terms of the average bus delay, its performance is not balanced across all traffic movements. The sensitivity analysis demonstrates that the proposed controller can effectively adapt to changes in bus occupancy and can still perform well even when the MPR is not 100%. Additionally, the controller's performance remains stable regardless of the bus arrival headway. In summary, the proposed DQN controller prioritizes buses while still maintaining a desirable level of service for other traffic. By ensuring transit vehicles receive the appropriate priority, our approach contributes to a more balanced transportation system. Moreover, its adaptability in handling changes in MPR and bus occupancy makes it a promising controller for real-world implementation. This study demonstrates the effectiveness of the proposed controller through comparison with conventional traffic signal control strategies. It is worth noting that numerous sophisticated algorithms have emerged in recent years, and a limitation of this study is that we did not assess the effectiveness of the controller in comparison to these algorithms, primarily due to constraints such as code availability.

The results also demonstrate that the PPO-based TSC controller using the multi-discrete action space and combined state space, exhibits superior performance in terms of both effectiveness and robustness. For effectiveness, it reduces average delays by 23.89% and 27.24% in peak and off-peak traffic demand conditions, respectively, when compared to the baseline (the pretimed signal controller). In terms of robustness, the average delays for both CVs and NCVs are lower than the baseline, even in scenarios with an MPR as low as 20%. Besides, the PPO algorithm can improve training efficiency compared to the DQN algorithm. In summary, PPO-based controllers outperform other controllers in a pure CV environment. The integration of the multi-discrete action space and the combined state space further enhances the performance of PPO-based controllers and mitigates its sensitivity to sparse observations. The findings highlight that the adoption of the PPO algorithm with multi-discrete actions and combined state space is a promising approach for real-world traffic signal control environments.

For the corridor scenarios, results indicate that MAPPO-M and MAPPO demonstrate the best performance in terms of bus metrics, while ASC and ATSP excel in car metrics. Regarding metrics related to the average person, ATSP demonstrates superior performance, and MARL-based controllers also perform well. MARL-based controllers demonstrate superior performance in hypothetical scenarios compared to real-world scenarios, probably attributed to the complexity disparities between these two types of scenarios. As the MRP increases, both MAPPO-M and MAPPO demonstrate improved performance in terms of average delays across all scenarios. This improvement becomes more pronounced when the MPR is lower than 60%. MAPPO in the hypothetical scenario with high traffic demand outperforms MAPPO-M in low MPR scenarios. However, MAPPO-M outperforms MAPPO in the other three scenarios when the MPR is low, providing more stable services for both buses and cars. This suggests that the introduction of multi-discrete action space can enhance the robustness of MARL-based controllers in mixed traffic environments, aligning with our findings in isolated intersection scenarios. When bus passenger occupancy increases from 1 to 30 passengers per bus, the average bus delay decreases significantly across all scenarios. When it exceeds 30, the MARL-based controllers become insensitive to it, especially during peak hours. The average car delay experiences a slight increase. However, the increases are minimal compared to the reduction in average bus delays, suggesting that the adverse impacts on the car are likely to be negligible, even with buses receiving more priority. Compared to MAPPO, except for the hypothetical scenario during peak hours with bus passenger occupancy less than 30, MAPPO-M provides better service for buses. Changes in bus arrival headway have insignificant impacts on the average bus delay, especially in scenarios controlled by MAPPO-M. As bus arrivals become less frequent, the average car delay decreases, along with the average person delay.

The traffic signal controllers proposed in this study addressed several limitations present in existing controllers, yet there are numerous opportunities for further improvement, particularly when considering the novelty of applying RL in the TSC domain. The following research directions are promising and worth attention:

1) **Integrating traffic state prediction components**: In mixed traffic environments with both CVs and non-CVs, the input state representation does not align with the assumption that it contains all relevant information for RL decision-making. Consequently, the performance of RL-based controllers becomes unreliable in such environments. Integrating traffic state prediction functions that use CVs' states to predict non-CVs' states can address this issue and significantly enhance the robustness of RL-based controllers.

2) **Developing hybrid controllers**: RL-based controllers make decisions within a black box, potentially leading to arbitrary actions that may be difficult for humans to interpret or trust in real-world implementations. To address this concern, developing hybrid controllers that combine the strengths of DRL algorithms with baseline control rules is a promising strategy. This integration can help avoid catastrophic behavior and ensure the reliability of the control system.

3) **Refining the basic RL model:** In this study, we employed vanilla RL algorithms without any modification. However, given the complexity of the TSP problem, effective customization of the RL model can significantly improve its performance. For example, incorporating recurrent neural networks (RNNs) to capture time-series features in consecutive inputs, integrating attention-based techniques to extract important information, etc., are avenues for refinement. Additionally, conducting more detailed investigations into the training and execution process is essential to pinpoint the theoretical limitations of the chosen RL algorithms and gain insight into how to refine them.

4) **Considering more comprehensive traffic conditions**: While the study evaluated the proposed controllers under various conditions, additional comprehensive evaluations are still worthy of being conducted. This includes scenarios with multiple transit priority requests, different bus stop locations, and oversaturated traffic flow situations. Additionally, traffic environments involve many kinds of traffic participants. While transit vehicles have been considered, pedestrians, bicycles, and motorcycles are also worth attention.

5) **Expanding the control objects**: This study primarily focused on optimizing traffic signals. With the advancement of autonomous vehicle (AV) technology, AVs are potential

control objects that can be integrated into the optimization process to achieve further progress. For instance, signal-vehicle coupled control is a promising research topic. In addition, applying MARL network-wide is also a challenging but promising research direction.

6) **Exploring different reward functions**: In exploring various optimization goals by constructing different reward functions, objectives may encompass mitigating bus bunching, stabilizing bus arrival headway, reducing greenhouse gas emissions, enhancing traffic safety, and more.

7) **Optimizing at the network level**: This study focuses solely on improving system efficiency at the operational level. Transit vehicles are granted conditional priorities based on their passenger occupancy, leading to fluctuations in bus arrival times due to uncertainty. These fluctuations can hinder the development of public transportation systems, as a stable arrival schedule facilitates trip planning and increases the attractiveness of public transportation. Future research could be conducted from the planning level. For example, researchers could solve network-level optimization problems involving transit vehicle priority, assuming a multimodal transportation environment with a fixed travel demand and flexible travel mode selection.

References

Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement learning for true adaptive traffic signal control. *J. Transp. Eng.*, *129*(3), 278-285.

Aslani, M., Mesgari, M. S., & Wiering, M. (2017). Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transp. Res. Part C Emerging Technol.*, *85*, 732-752. https://doi.org/10.1016/j.trc.2017.09.020

Chow, A. H. F., Su, Z. C., Liang, E. M., & Zhong, R. X. (2021). Adaptive signal control for bus service reliability with connected vehicle technology via reinforcement learning. *Transp. Res. Part C Emerging Technol.*, *129*, 103264. https://doi.org/10.1016/j.trc.2021.103264

Christofa, E., & Skabardonis, A. (2011). Traffic Signal Optimization with Application of Transit Signal Priority to an Isolated Intersection. *Transp. Res. Rec.*, *2259*(1), 192-201. https://doi.org/10.3141/2259-18

Dongbin, Z., Yujie, D., & Zhen, Z. (2012). 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. https://doi.org/10.1109/tsmcc.2011.2161577

El-Tantawy, S., & Abdulhai, B. (2010). An agent-based learning towards decentralized and coordinated traffic signal control. *13th International IEEE Conference on Intelligent Transportation Systems*, 665-670.

Erdmann, J. (2015, 2015//). SUMO's Lane-Changing Model. *Modeling Mobility with Open Data*, 105-123.

Feng, Y., Head, K. L., Khoshmagham, S., & Zamanipour, M. (2015). A real-time adaptive signal control in a connected vehicle environment. *Transp. Res. Part C Emerging Technol.*, *55*, 460-473. https://doi.org/10.1016/j.trc.2015.01.007

García-Nieto, J., Alba, E., & Carolina Olivera, A. (2012). Swarm intelligence for traffic light scheduling: Application to real urban areas. *Eng. Appl. Artif. Intell.*, *25*(2), 274-283. https://doi.org/10.1016/j.engappai.2011.04.011

Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142*.

Ghanim, M. S., & Abu-Lebdeh, G. (2015). Real-Time Dynamic Transit Signal Priority Optimization for Coordinated Traffic Networks Using Genetic Algorithms and Artificial Neural Networks. *J. Intell. Transp. Syst.*, *19*(4), 327-338. https://doi.org/10.1080/15472450.2014.936292

Guo, Q., Li, L., & Ban, X. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transp. Res. Part C Emerging Technol.*, *101*, 313-334. https://doi.org/10.1016/j.trc.2019.01.026

Haydari, A., & Yilmaz, Y. (2022). Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.*, *23*(1), 11-32. https://doi.org/10.1109/tits.2020.3008612

He, Q., Head, K. L., & Ding, J. (2014). Multi-modal traffic signal control with priority, signal actuation and coordination. *Transp. Res. Part C Emerging Technol.*, *46*, 65-82. https://doi.org/10.1016/j.trc.2014.05.001

Koonce, P. (2008). *Traffic signal timing manual* (No. FHWA-HOP-08-024). United States. Federal Highway Administration. https://rosap.ntl.bts.gov/view/dot/20661

Lee, J., Abdulhai, B., Shalaby, A., & Chung, E.-H. (2006). Real-Time Optimization for Adaptive Traffic Signal Control Using Genetic Algorithms. *J. Intell. Transp. Syst.*, *9*(3), 111-122. https://doi.org/10.1080/15472450500183649

Li, L., Lv, Y., & Wang, F.-Y. (2016). Traffic signal timing via deep reinforcement learning. *IEEE/CAA J. Autom. Sin.*, *3*(3), 247-254.

Li, W., & Ban, X. (2019). Connected Vehicles Based Traffic Signal Timing Optimization. *IEEE Trans. Intell. Transp. Syst.*, *20*(12), 4354-4366. https://doi.org/10.1109/tits.2018.2883572

Liang, X., Du, X., Wang, G., & Han, Z. (2019). A Deep Reinforcement Learning Network for Traffic Light Cycle Control. *IEEE Trans. Veh. Technol.*, *68*(2), 1243-1253. https://doi.org/10.1109/tvt.2018.2890726

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, *30*.

Ma, W., Yang, X., & Liu, Y. (2010). Development and Evaluation of a Coordinated and Conditional Bus Priority Approach. *Transp. Res. Rec.*, *2145*(1), 49-58. https://doi.org/10.3141/2145-06

Mannion, P., Duggan, J., & Howley, E. (2016). An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control. In T. L. McCluskey, A. Kotsialos, J. P. Müller, F. Klügl, O. Rana, & R. Schumann (Eds.), *Autonomic Road Transport Support Systems* (pp. 47-66). Springer International Publishing. https://doi.org/10.1007/978-3-319-25808-9_4

Mao, F., Li, Z., Lin, Y., & Li, L. (2023). Mastering Arterial Traffic Signal Control With Multi-Agent Attention-Based Soft Actor-Critic Model. *IEEE Trans. Intell. Transp. Syst.*, *24*(3), 3129-3144. https://doi.org/10.1109/tits.2022.3229477

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529-533. https://doi.org/10.1038/nature14236

Mohamad Alizadeh Shabestary, S. (2019). *Deep Reinforcement Learning Approach to Multimodal Adaptive Traffic Signal Control* (Publication Number 13422652) [Ph.D., University of Toronto (Canada)]. ProQuest Dissertations & Theses Global. Ann Arbor.

Priemer, C., & Friedrich, B. (2009). A decentralized adaptive traffic signal control using V2I communication data. *2009 12th international ieee conference on intelligent transportation systems*, 1-6.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. *International conference on machine learning*, 1889-1897.

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shabestary, S. M. A., & Abdulhai, B. (2022). Adaptive Traffic Signal Control With Deep Reinforcement Learning and High Dimensional Sensory Inputs: Case Study and Comprehensive Sensitivity Analyses. *IEEE Trans. Intell. Transp. Syst.*, 1-15. https://doi.org/10.1109/tits.2022.3179893

Skabardonis, A. (2000). Control Strategies for Transit Priority. *Transp. Res. Rec.*, *1727*(1), 20-26. https://doi.org/10.3141/1727-03

Skabardonis, A., & Geroliminis, N. (2008). Real-Time Monitoring and Control on Signalized Arterials. *J. Intell. Transp. Syst.*, *12*(2), 64-74. https://doi.org/10.1080/15472450802023337

Song, L., & Fan, W. D. (2023). Performance of State-Shared Multiagent Deep Reinforcement Learning Controlled Signal Corridor with Platooning-Based CAVs. *Journal of Transportation Engineering, Part A: Systems*, *149*(8). https://doi.org/10.1061/jtepbs.Teeng-7768

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Teklu, F., Sumalee, A., & Watling, D. (2007). A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer‐Aided Civil and Infrastructure Engineering*, *22*(1), 31-43.

Thorpe, T. L., & Anderson, C. W. (1996). Traffic light control using SARSA with three state representations. *Technical report, Citeseer*.

Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, *62*(2), 1805-1824. https://doi.org/10.1103/physreve.62.1805

U.S. Department of Transportation. (2011). *High-Priority Applications and Development Approach*. Retrieved February 15, 2023 from https://www.its.dot.gov/press/2011/mobility_app.htm

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. *Proceedings of the AAAI conference on artificial intelligence*, 2094-2100.

Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transp. Res. Part C Emerging Technol.*, *36*, 177-195.

Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., & Li, Z. (2019). PressLight: Learning max pressure control to coordinate traffic signals in arterial network. *In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1290-1298. https://doi.org/10.1145/3292500.3330949

Wei, H., Zheng, G. J., Yao, H. X., & Li, Z. H. (2018, Aug 19-23). IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. *In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2496-2505. https://doi.org/10.1145/3219819.3220096

Wunderlich, R., Liu, C., Elhanany, I., & Urbanik, T. (2008). A Novel Signal-Scheduling Algorithm With Quality-of-Service Provisioning for an Isolated Intersection. *IEEE Trans. Intell. Transp. Syst.*, *9*(3), 536-547. https://doi.org/10.1109/tits.2008.928266

Yang, T., & Fan, W. (2023). Evaluation of transit signal priority at signalized intersections under connected vehicle environment. *Transp. Plann. Technol.*, 1-15. https://doi.org/10.1080/03081060.2023.2176308

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, *35*, 24611-24624.

Yu, J., Laharotte, P.-A., Han, Y., & Leclercq, L. (2023). Decentralized signal control for multi-modal traffic network: A deep reinforcement learning approach. *Transp. Res. Part C Emerging Technol.*, *154*. https://doi.org/10.1016/j.trc.2023.104281

Zeng, X., Zhang, Y., Jiao, J., & Yin, K. (2021). Route-Based Transit Signal Priority Using Connected Vehicle Technology to Promote Bus Schedule Adherence. *IEEE Trans. Intell. Transp. Syst.*, *22*(2), 1174-1184. https://doi.org/10.1109/tits.2020.2963839

Zhang, R., Ishikawa, A., Wang, W., Striner, B., & Tonguz, O. K. (2021). Using Reinforcement Learning With Partial Vehicle Detection for Intelligent Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.*, *22*(1), 404-415. https://doi.org/10.1109/tits.2019.2958859