ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Intelligent Traffic Intersection Management through Multi-Agent Reinforcement Learning

TOMÁS ALEXANDRE HENRIQUES ANTUNES

(Licenciado em Engenharia Eletrónica e Telecomunicações e de Computadores)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e Multimédia

Orientadores: Doutora Maria Manuela Almeida Carvalho Vieira
Doutor Mário Pereira Véstias
Doutora Paula Maria Garcia Louro

Júri:

Presidente: Doutor Diogo Nuno Crespo Ribeiro Cabral
Vogais: Doutor José Manuel Matos Ribeiro da Fonseca
Doutor Mário Pereira Véstias

Dezembro 2025

Intelligent Traffic Intersection Management through Multi-Agent Reinforcement Learning

TOMÁS ALEXANDRE HENRIQUES ANTUNES

(Licenciado em Engenharia Eletrónica e Telecomunicações e de Computadores)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e Multimédia

Orientadores: Doutora Maria Manuela Almeida Carvalho Vieira, ISEL
Doutor Mário Pereira Véstias, ISEL
Doutora Paula Maria Garcia Louro, ISEL

Júri:

Presidente: Doutor Diogo Nuno Crespo Ribeiro Cabral, ISEL
Vogais: Doutor José Manuel Matos Ribeiro da Fonseca, FCT - UNL
Doutor Mário Pereira Véstias, ISEL

Dezembro 2025

Acknowledgements

The development of this project was both a demanding and enriching journey, marked by challenges as well as countless opportunities for growth. This dissertation would not have been possible without the guidance, support, and encouragement of many people. I am deeply grateful to all those who contributed in different ways to the successful completion of this work.

First, I express my gratitude to the professors Manuela Vieira, Paula Louro, and Mário Véstias, who supervised me during this period. Their availability, advice and constant support were invaluable, and I am grateful for the opportunity to have worked under their guidance.

I would also like to extend my thanks to Professor Manuel Vieira, who, although not my supervisor, was always present in our meetings and played an essential role in the progress of this project. I am also especially grateful to my colleague Gonçalo Galvão, whose collaboration and contribution were a key element in overcoming challenges and achieving results.

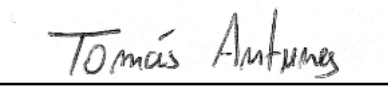
I am thankful to ISEL, the institution that has welcomed me over the past years, not only for supporting my academic development but also for everything it has provided me during this journey. Beyond the classroom, ISEL offered me experiences, friends and opportunities that have shaped both my professional and personal growth.

Finally, I wish to thank my family for their love and support. I am also grateful to my cat, who was always by my side during the long hours at my desk, reminding me to take breaks. To my parents, for their endless encouragement and the many sacrifices they made so I could pursue my studies, this work is dedicated to you. And to you, Inês, thank you for your patience, love, and for being always by my side.

Statement of integrity

I declare that this dissertation is the result of my personal and independent research. Its content is original, and all sources listed in the bibliographic references were consulted and are duly mentioned in the text. I further declare that all scientific and technical references relevant to the development of the work are duly cited and included in the bibliographic references.

The author



Lisbon, December 17, 2025

Abstract

Urban traffic management remains a persistent challenge for modern cities, particularly during peak hours when large volumes of vehicles and pedestrians converge, causing severe congestion, delays, and increased road safety risks. Given the limited feasibility of expanding physical infrastructure, it becomes essential to explore intelligent and adaptive solutions for traffic signal control.

This work focuses on the application of Multi-Agent Reinforcement Learning (MARL) algorithms to optimize decision-making and coordination of traffic signals in urban networks. It is assumed that Visible Light Communication (VLC) between vehicles and infrastructure is available to provide real-time data required for the decision process, although this component is not the primary focus of the research.

To validate the proposed approach, a simulation environment was developed using *Simulation of Urban Mobility* (SUMO), consisting of five interconnected signalized intersections. Within this context, different MARL algorithms were studied and compared, including *Deep Q-Learning Network* (DQN) and *Multi-Agent Proximal Policy Optimization* (MAPPO), with the objective of evaluating their performance under heterogeneous and dynamic traffic scenarios.

The results show that MAPPO consistently outperforms DQN-based methods, achieving faster and more complete clearance of vehicles while maintaining lower waiting times for pedestrians. QT-DQN provides slight improvements over DQN in vehicle flow but at the cost of harming pedestrian performance. Overall, the study demonstrates that MARL methods, and particularly MAPPO, offer significant improvements in traffic efficiency and fairness, reinforcing their potential for deployment in real-world urban environments.

Keywords: Multi-Agent Reinforcement Learning, Visible Light Communication, Artificial Intelligence, Traffic Signal Control, Urban Traffic Control, Road Safety.

Resumo

A gestão do tráfego urbano constitui um desafio persistente para as cidades modernas, sobretudo durante as horas de ponta, quando grandes volumes de veículos e peões convergem, originando congestionamentos, atrasos e riscos acrescidos de segurança rodoviária. Face à limitada viabilidade da expansão da infraestrutura física, torna-se essencial explorar soluções inteligentes e adaptativas para o controlo da sinalização semafórica.

Este trabalho foca-se na aplicação de algoritmos de Aprendizagem por Reforço Multiagente (MARL) para otimizar a decisão e coordenação dos semáforos em redes urbanas. Assume-se a existência de Comunicação por Luz Visível (VLC) entre veículos e infraestrutura, utilizada para disponibilizar em tempo real os dados necessários ao processo de decisão, embora esta componente não constitua o foco principal da investigação.

Para a validação da proposta, foi desenvolvido um ambiente de simulação baseado no simulador *Simulation of Urban Mobility* (SUMO), composto por cinco interseções com semáforos interligadas. Neste contexto, foram estudados e comparados diferentes algoritmos de MARL, incluindo o *Deep Q-Learning Network* (DQN) e o *Multi-Agent Proximal Policy Optimization* (MAPPO), com o objetivo de avaliar o seu desempenho em cenários de tráfego heterogéneo e dinâmico.

Os resultados mostram que o MAPPO supera consistentemente os métodos baseados em DQN, alcançando uma escoação mais rápida e completa dos veículos, ao mesmo tempo que mantém menores tempos de espera para os peões. O QT-DQN apresenta ligeiras melhorias em relação ao DQN no escoamento dos veículos, mas com impacto negativo no desempenho pedonal. De forma geral, o estudo demonstra que os métodos MARL, em particular o MAPPO, oferecem melhorias significativas na eficiência e justiça do tráfego, reforçando o seu potencial para aplicação em ambientes urbanos reais.

Palavras-chave: Aprendizagem por Reforço Multiagente, Comunicação por Luz Visível, Inteligência Artificial, Controlo da Sinalização Semafórica, Gestão de Tráfego Urbano, Segurança Rodoviária.

Contents

List of Figures

List of Tables

Listings

Acronyms

AI	Artificial Intelligence 1
ATSC	Adaptive Traffic Signal Control 6
CAVs	Connected and Autonomous Vehicles 14
CV	Connected Vehicle 14
C-V2X	Cellular Vehicle-to-Everything 13
DQN	Deep Q-Learning Network 2
DSRC	Dedicated Short-Range Communication 13
GNN	Graph Neural Network 11
I2P	Infrastructure-to-Pedestrian 15
I2V	Infrastructure-to-Vehicle Communication 14
IM	Intersection Manager 18
ITS	Intelligent Transportation Systems 12
LED	Light Emitting Diode 15
Li-Fi	Light Fidelity 15
MAPPO	Multi-Agent Proximal Policy Optimization 2
MARL	Multi-Agent Reinforcement Learning 2
MSE	Mean Squared Error 27
NN	Neural Network 9
OHE	One-Hot Encoding 34
P2I	Pedestrian-to-Infrastructure 15
PPO	Proximal Policy Optimization 10
QT-CDQN	Cooperative Deep Q-Learning Network with Q-value transfer 30
ReLU	Rectified Linear Unit 28

RF	Radio Frequency 14
RL	Reinforcement Learning 1
SUMO	Simulation of Urban MObility 14
TraCI	Traffic Control Interface 17
V2I	Vehicle-to-Infrastructure Communication 13
V2V	Vehicle-to-Vehicle Communication 13
VLC	Visible Light Communication 2
V-VLC	Vehicular Visible Light Communication 15

1 Introduction

This first chapter serves as the introduction to the document, where the motivations for writing it, the proposed objectives, how it is organized are revealed and the publications that arose from it.

1.1 Motivation

Urban mobility has become one of the most pressing challenges for modern cities. Efforts to improve it often include strengthening public transportation systems, encouraging the use of sustainable modes of travel, and implementing policies to reduce car dependency. However, even with such measures, urban traffic congestion remains a persistent problem. During peak hours, the convergence of high volumes of vehicles and pedestrians leads to long delays, excessive queueing, and an increased risk of accidents. Since expanding physical infrastructure is rarely a viable option due to financial, spatial, and environmental constraints, it becomes essential to explore alternative strategies that optimize the use of existing road networks.

Traditional traffic signal control systems, often based on fixed-time schedules, are limited in their ability to adapt to rapidly changing traffic conditions. As a result, they struggle to provide efficient traffic flow and to ensure road safety in complex urban environments.

In this context, intelligent and adaptive traffic management emerges as a promising approach. By leveraging the potential of Artificial Intelligence (AI), and in particular Reinforcement Learning (RL), traffic signals can move beyond static or pre-programmed logic towards systems capable of making real-time decisions. For example, such systems could dynamically select the most appropriate signal phase at a given moment or adjust the duration of green and red phases according to the actual traffic demand. This paradigm has the potential not only to reduce congestion and travel times but also to improve safety for both drivers and pedestrians while contributing to lower pollution levels in urban areas.

1.2 Objectives

The main objective of this work is to design and evaluate AI-based approaches for traffic signal control at urban intersections using Multi-Agent Reinforcement Learning (MARL), with the goal of enhancing the efficiency, adaptability, and coordination of urban traffic management.

To achieve these goals, a simulation environment with multiple interconnected intersections will be implemented, assuming the availability of real-time data through Visible Light Communication (VLC). Within this framework, reinforcement learning algorithms - Deep Q-Learning Network (DQN) and Multi-Agent Proximal Policy Optimization (MAPPO) - will be developed, trained, and compared. By addressing the growing challenges of congestion and road safety, the work aims to contribute to the development of scalable and intelligent mobility solutions for modern cities. The ultimate objective is to reduce average waiting times, queue lengths, and overall traffic congestion, thereby enhancing safety for both drivers and pedestrians while contributing to lower levels of urban pollution.

1.3 Document Outline

This document is organized into six chapters. Chapter 1 introduces the motivation, objectives, and structure of the work. Chapter 2 reviews the state of the art in traffic management, reinforcement learning, and vehicular communication. Chapter 3 presents the simulation environment and traffic scenarios, while Chapter 4 details the implemented algorithms. Chapter 5 discusses the results, and Chapter 6 concludes the work with final remarks and future work.

1.4 Resulting Outputs

This Master's Thesis was developed within the framework of the project IPL/IDI&CA2024/INUTRAM_ISEL. The activities developed in this context resulted in the following outputs:

1. T. Antunes, G. Galvão, M. A. Vieira, M. Vieira, M. Véstias, and P. Louro, "Intelligent Intersection Management through Multi-Agent Reinforcement Learning and Visible Light Communication Integration," Proceedings of the 11th International Conference on Sensors and Eletronic Intrumentation Advances (SEIA' 2025), 24-26 September 2025, Ponta Delgada, São Miguel, Azores, Portugal.

2. M. A. Vieira, T. Antunes, G. Galvão, M. Vieira, M. Véstias, and P. Louro, "Intelligent Intersection Management through Multi-Agent Reinforcement Learning, Self-Adaptive Phase Adjustment and Visible Light Communication", Sensors & Transducers, vol. 270, no. 3, pp. 69–78, Nov. 2025.

3. T. Antunes, G. Galvão, M. A. Vieira, M. Vieira, M. Véstias, and P. Louro, "Decentralized Smart Traffic Signal Control Using IoT-Based Multi-Agent Reinforcement

Learning and VLC Communication," Proceedings of SPIE Photonics West 2026, San Francisco, CA, USA, 2026.

2 State of the Art

2.1 Traffic Management

Transport plays a central role in daily life, and the increasing reliance on it has led to a rise in the number of vehicles on the roads, which leads to an increase in traffic congestion. This scenario is evident during peak hours, such as morning and evening commutes, when most of the people head towards their workplaces or back home. A traffic control system manages the ?? of signalization for both vehicles and pedestrians. Some of the main causes of conflict involving pedestrians and vehicles are due to poor control of phases and their activation times, which means good control of pedestrian movement on the roads is extremely important [1].

The fixed-time traffic control system is still the most commonly adopted system; however, it fails to ensure optimal waiting times on the roads [2]. Because it operates in a non-adaptive way, it frequently generates congestion when traffic flow is unbalanced. This type of system does not include mechanisms to estimate or respond to current traffic demand. Consequently, even when traffic densities vary, the signal phases remain fixed, giving equal time to both lightly and heavily loaded roads. This results in inefficient use of time and contributes to longer overall travel durations.

Long unnecessarily waiting times for the green light for pedestrians, even if there are no vehicles on the road being served, can lead to a lack of patience on the part of pedestrians to wait, leading them to cross at a red light. In cases of large concentrations of people in waiting areas, due to the fixed green time for the phases without adapting to the density of pedestrians, this time can be too short for everyone to pass safely within the time limit, causing some to cross when the phase is no longer active, which reduces pedestrian safety [3].

To address these challenges, more intelligent traffic signal control systems are required, capable of dynamically adjusting signal phases in response to real-time traffic and pedestrian demand.

2.1.1 Conventional Traffic Signal Systems

Conventional traffic signal systems can be broadly divided into three main categories. The first is the *fixed-time* control, in which signal phases follow pre-defined cycles with fixed durations. This approach is simple and effective when traffic patterns are stable and predictable, but it lacks the flexibility to handle variations in flow or unexpected events.

A second category is the *actuated control*, where the duration of signal phases is adjusted in real time based on input from local detectors, such as vehicle presence sensors or pedestrian push buttons. This allows greater responsiveness to local demand compared to fixed-time systems, although the scope of adaptation is limited and coordination between multiple intersections remains a challenge.

Finally, there are approaches based on *model predictive control (MPC)*, which use mathematical models of traffic dynamics to forecast conditions and optimize signal timings accordingly. While MPC offers a more systematic way of adapting to changing flows, its effectiveness in practice is often constrained by model accuracy and computational requirements.

According to [4], fixed-time systems perform reasonably well under stable traffic, while actuated control shows advantages when flows fluctuate or unexpected events occur. However, both methods face limitations in scalability and inter-intersection coordination, reinforcing the need for more Adaptive Traffic Signal Control (ATSC) strategies.

Beyond this review, several studies have examined MPC in comparison with fixed-time and actuated controllers. In [5], it is reported that MPC significantly reduces queue lengths and outperforms fixed-time control in a single-intersection case study in Tehran. MPC has also been combined with neural networks, showing that while improvements are modest under low traffic demand, MPC-based strategies yield substantial gains in average speed and queue reduction under heavy traffic conditions compared to both fixed-time and sensor-based actuated methods [6].

2.1.2 Adaptive Traffic Signal Control

To overcome these limitations, artificial intelligence offers a promising approach. AI-based adaptive traffic signal control can dynamically determine which traffic light phase to activate, optimize the duration of green and red lights, and even prioritize directions according to traffic demand throughout the day. With this implementation, it would be possible to improve the system and traffic flow on the roads [7].

When compared with actuated control and fixed-time plans, recent learning-based AI approaches have shown clear advantages. In one study, an adaptive method was tested in a single-intersection scenario using simulated data [8]. The results indicated that this controller reduced average delay by approximately 32% relative to actuated control and by about 37% compared to fixed-time operation. These findings suggest that machine learning techniques have the potential to substantially outperform conventional systems, even under controlled experimental settings.

This topic has been extensively studied as a means to improve urban mobility by dynamically adjusting signal timings in response to traffic demand. While these methods can reduce delays under certain conditions, their reliance on predefined rules and coordination might limit its adaptability in highly dynamic or heterogeneous traffic environments. Furthermore, scalability can become problematic when extending such systems to large urban networks with diverse mobility patterns [9].

One of the acceptations of the goals of AI is to develop machines that resemble the intelligent behaviour of a human being. In order to achieve this goal, an AI system should be able to interact with the environment and learn how to correctly act inside it. An established area of AI that has been proved capable of dealing very well with this problem is reinforcement learning.

2.2 Multi-Agent Reinforcement Learning

Reinforcement Learning (RL) has emerged as a promising paradigm for traffic signal control, enabling agents to learn optimal policies through interaction with the environment. Early studies focused on single-agent RL, demonstrating improvements in local intersection performance [10]. However, as stated earlier, centralized RL approaches usually struggle to scale to city-wide networks due to computational complexity and communication overhead.

Multi-Agent Reinforcement Learning (MARL) has therefore gained traction, allowing distributed agents to coordinate locally while addressing scalability challenges. MARL extends single-intersection reinforcement learning control to a stochastic game environment, where multiple agents (intersections) interact simultaneously across arterial or regional traffic networks [11] [12].

The following figure illustrates a simple reinforcement learning scheme with multiple agents, where each agent represents a traffic intersection.

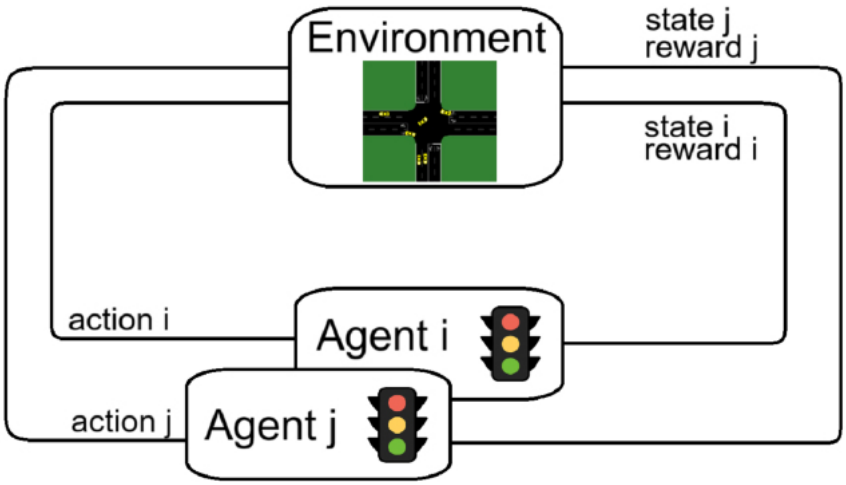


Figure 2.1: *Schematic representation of a multi-agent reinforcement learning framework for traffic signal control.*

The aim is to reach a global equilibrium that enhances overall traffic efficiency. A key difficulty, however, is that agents update their behaviors concurrently, the goal of each

agent becomes non-stationary, as it is continuously influenced by others. In practice, the optimal strategy at one intersection is continuously affected by the evolving behaviors of neighboring intersections.

Most applications of MARL to traffic signal control focus on efficiency-related indicators, such as reducing travel times or vehicle waiting times. However, equally important aspects, such as road safety and fairness between different traffic participants, have received comparatively little attention, limiting their robustness in real-world applications. To overcome these limitations, more recent studies have proposed hybrid or hierarchical MARL frameworks, where a central agent learns global coordination strategies while individual intersections act as local agents, applying actions optimized according to multiple reward signals that balance efficiency with safety objectives. These architectures show promise in enhancing scalability, effective cooperation, coordination and robustness in complex urban environments.

MARL methods can be further categorized into *value-based* and *policy-based* methods. Value-based methods, such as *Deep Q-Learning, SARSA* or *QMIX*, learn a value function to estimate the expected return of each possible action, selecting the action with the highest value. These methods are often favored for their simplicity and computational efficiency, and have been widely applied to traffic control scenarios, particularly in independent or parameter-sharing settings [13].

Policy-based methods, on the other hand, learn an explicit policy representation and attempt to improve the policy in the direction of the gradient, by learning a parameterized mapping from states to actions. They are particularly effective in continuous or large action spaces and for promoting cooperative behaviors among agents [14]. Most of the algorithms in this family adopt an actor-critic architecture: the actor is responsible for selecting actions according to the learned policy, while the critic evaluates those actions by estimating their expected return. This interaction reduces variance during training and enables richer evaluation of action dependencies. Algorithms such as *MAPPO*, *MADDPG* or *COMA* fall into this category and have been successfully applied to traffic management when stronger coordination among intersections is required [15].

In practice, while value-based algorithms remain popular for their efficiency, policy-based methods are often reported to achieve superior performance in large-scale urban traffic networks, largely because their ability to optimize policies directly makes them more robust to the non-stationary dynamics that arise when multiple agents learn and adapt simultaneously. Lastly, a particular advantage of actor–critic policy-based methods is that the critic can evaluate the quality of actions not only independently, but also by using policy gradients on joint observations and actions, enabling global coordination across agents [16]. This enables the learning of effective phase coordination strategies across neighboring intersections, allowing policies to capture interdependencies more effectively and adapt better than value-based methods, which typically treat each phase as an isolated action without modeling such interactions.

Table 2.1: Comparison of traffic signal control approaches

Method	Examples	Advantages	Limitations
Fixed-time control	Predefined signal plans with static cycles	Simple to implement; predictable	Not adaptive; poor performance under variable or unexpected traffic
Actuated control	Sensors (e.g., speed sensors, inductive loops, pedestrian buttons) trigger changes	Reacts to real-time traffic/-pedestrian demand; more adaptive than fixed-time	Local adaptation only; high installation and maintenance cost;
Model Predictive Control (MPC)	Optimization using traffic flow models	Anticipates future conditions; can improve global performance	High computational cost; scalability issues in large networks
Value-based MARL	Q-Learning, SARSA, QMIX	Computationally efficient; often easier to implement and understand; performs well in environments with a finite set of actions	Limited coordination; may struggle in continuous or non-stationary environments
Policy-based MARL	MAPPO, MADDPG, COMA	Handles continuous/large action spaces; promotes cooperative behavior; better robustness in non-stationary settings; actor-critic models can capture interdependencies between agents	Generally requires more data to converge; often more complex to implement and tune; policy gradient estimates can have high variance;

2.2.1 Deep Q-Learning

Deep Q-Learning (DQN) has become one of the most used value-based reinforcement learning methods for traffic signal control. Unlike tabular Q-Learning, which stores values for each state-action pair and is limited to small environments, DQN uses deep Neural Network (NN) to approximate the action-value function $Q(s, a)$. This allows agents to operate in complex, high-dimensional environments such as urban traffic networks, where the number of possible states and actions grows exponentially.

Several works have reported that DQN-based agents can significantly reduce average travel times, shorten vehicle queues, and improve throughput compared to fixed-time or actuated controllers [17]. A central aspect of these approaches lies in the design of the reward function. For example, a common formulation is to define the reward as the decrease in total waiting time between consecutive steps, encouraging continuous improvement in traffic flow. Such formulations enable agents to prioritize not only efficiency but also safety and fairness across different road users.

In multi-agent traffic signal control, DQNs are often deployed with one agent per intersection. To address scalability challenges and promote cooperation, parameter sharing is commonly applied, where a single network is trained and used across multiple agents. Additionally,

neighborhood information can be integrated into the state representation, allowing each agent to consider the conditions of adjacent intersections when making decisions. These mechanisms balance local optimization with global coordination, improving performance across larger traffic networks [18].

In [19] it is proposed a DQN with Q-value transfer, where agents incorporate the optimal Q-values of neighboring intersections into their own loss function. This mechanism improves coordination by allowing agents to consider the influence of recent neighbor actions during policy learning. The results show that this approach is competitive in terms of different metrics when compared to standard DQN.

In the traffic signal control problem, the observable state typically includes traffic-related features around the intersection, such as queue lengths, waiting times, velocity, or the presence of pedestrians. The agent's actions correspond to changes in signal phases or the adjustment of phase durations, while the reward function is designed to reflect traffic efficiency - often measured by reductions in total waiting time, queue lengths, or delay across time steps [20]. By maximizing cumulative rewards, DQN agents learn policies that minimize congestion and improve overall mobility.

Despite these advantages, DQN has notable limitations. Previous studies have shown that DQN, as an off-policy algorithm, suffers from instability in non-stationary multi-agent environments [21]. For instance, independent DQN agents provide no theoretical guarantees of convergence when neighboring agents update their policies concurrently, which violates the Markov stationarity assumption of the decision process [22]. Furthermore, parameter sharing, while efficient, constrains the ability to learn highly specialized policies for heterogeneous intersections. These challenges limit the scalability of DQN to large, diverse urban networks, motivating the exploration of alternative policy-based methods such as MAPPO.

2.2.2 Multi Agent Proximal Policy Optimization

Proximal Policy Optimization (PPO) [23] is a widely used policy-gradient algorithm in reinforcement learning that addresses the instability often observed in policy optimization. PPO improves training stability by introducing a clipped surrogate objective, which constrains large policy updates while still allowing sufficient exploration [24]. This balance between exploration and stability has made PPO a strong baseline in many reinforcement learning domains [25].

MAPPO extends PPO to the multi-agent setting, following the paradigm of centralized training with decentralized execution (CTDE). In this framework, each agent learns a decentralized policy (actor), while a centralized critic evaluates the joint observations and actions of all agents during training. This design enables MAPPO to capture dependencies across agents and promote coordinated behavior, while ensuring that execution remains scalable and distributed.

Compared to value-based methods such as DQN, MAPPO provides several advantages in the traffic signal control domain. First, by directly optimizing parameterized policies,

MAPPO is better suited to handle large or continuous action spaces, which can represent different signal phase durations or adaptive timings. Second, the centralized critic incorporates information from neighboring intersections, allowing the algorithm to capture interdependencies across traffic lights, an aspect where independent DQN agents often struggle. Finally, PPO's clipped objective improves training stability and reduces the risk of divergence, a limitation commonly encountered in Q-learning when applied to dynamic, non-stationary environments [25].

The MAPPO algorithm can be applied to traffic signal control by modeling each intersection as an independent agent responsible for selecting signal phases. During training, the centralized critic evaluates the joint state of the traffic network, including vehicle queues and pedestrian flows, while each agent updates its policy based on this feedback. This allows agents to coordinate decisions, reducing the likelihood of conflicting signal changes between adjacent intersections. In [26] it was demonstrated a similar actor–critic approach for traffic control, showing that multi-agent policy-gradient methods can outperform DQN in terms of delay reduction and traffic throughput.

In summary, MAPPO builds upon PPO's stability and scalability while leveraging centralized critic to promote coordination between agents. These features make MAPPO particularly well suited to traffic signal control, where multiple intersections must adapt simultaneously to highly dynamic and interdependent traffic conditions.

2.3 Graph Neural Network

Graph Neural Network (GNN) is a family of deep learning models specifically designed to process graph-structured data, where nodes represent entities and edges represent relationships between them. Unlike conventional neural networks that assume fixed-size vector inputs, GNNs exploit the topology of graphs to capture both local and global dependencies through iterative message passing between nodes [27].

In the context of traffic signal control, urban road networks can be naturally modeled as graphs, where intersections correspond to nodes and connecting roads to edges. Each node (intersection) processes local traffic states such as vehicle queues, waiting times, or pedestrian demand, while also receiving information from adjacent nodes (neighboring intersections). By propagating these features across the graph, a GNN learns spatial-temporal representations that capture both local conditions and global traffic dynamics. This structure enables more informed and coordinated decision-making compared to approaches that treat intersections independently.

Compared with traditional MARL-based approaches such as DQN or MAPPO, GNNs offer some distinctive advantages. First, they provide an efficient and scalable way to encode interdependencies between intersections without requiring explicit centralized critics. Second, and very importantly, they can generalize to different network topologies, making them suitable for deployment in varying city layouts.

Furthermore, the relatively low interpretability of GNNs remains a barrier for their adoption in decision-making contexts, where transparency and explainability are critical for policy

evaluation. Addressing these limitations is essential to fully exploit the potential of GNN-based approaches in Intelligent Transportation Systems (ITS) [28].

Recent surveys confirm the growing role of GNNs in Intelligent Transportation Systems (ITS). In [28], it is shown that GNNs are particularly effective in capturing complex spatio-temporal dependencies in non-Euclidean networks, such as urban road systems. Most existing research focuses on traffic forecasting tasks (e.g., predicting vehicle flow, speed, or density), while emerging applications extend to traffic signal control, safety, and urban planning. Despite their promising performance, several challenges still limit the deployment of GNNs in real-world traffic management. Key issues include the limited generalizability of current models to unseen or rare scenarios, the need to ensure computational efficiency on large-scale graphs under real-time constraints, and the difficulty of handling heterogeneous and noisy sensor data.

In summary, GNNs provide a powerful alternative to value-based and policy-based MARL approaches by leveraging the natural graph structure of urban road networks. Their ability to model spatial dependencies makes them particularly attractive for large-scale traffic management problems, although further work is required to address their computational cost and robustness in real-world deployments.

Table 2.2: Comparison of DQN, MAPPO, and GNN approaches for traffic signal control.

Method	Characteristics	Advantages	Limitations
Deep Q-Learning (DQN)	Value-based method that approximates $Q(s, a)$ with a neural network. Usually trained independently or with parameter sharing across intersections.	• Simple and efficient in small/medium-scale problems • Effective in reducing waiting times and queues • Well-studied and widely applied	• Struggles with non-stationary environments • Limited coordination between intersections • Scalability issues with heterogeneous networks
MAPPO	Policy-based actor–critic method. Centralized training with decentralized execution (CTDE). Uses a centralized critic and decentralized actors.	• Better stability and robustness than DQN • Captures interdependencies via centralized critic • Handles continuous and large action spaces	• Higher computational cost • Requires more training data • Still sensitive to reward design
Graph Neural Networks (GNNs)	Neural models designed for graph-structured data. Intersections = nodes, roads = edges. Coordination via message passing between neighbors.	• Naturally models spatial dependencies • Scalable to arbitrary network topologies • Coordination emerges from graph structure	• Computationally expensive for large graphs • Sensitive to graph representation design • Interpretability and fairness underexplored

2.4 Vehicular Communication Technologies

Communication technologies are essential to the development of Intelligent Transportation Systems, acting as the foundation for exchanging information between vehicles, roadside infrastructure, and central control units. Conventional solutions such as Dedicated Short-Range Communication (DSRC) and Cellular Vehicle-to-Everything (C-V2X) have been extensively explored to enable Vehicle-to-Vehicle Communication (V2V) and Vehicle-to-Infrastructure Communication (V2I) interactions. Despite their maturity, these approaches still face persistent obstacles, including latency, interference, spectrum scarcity, and high deployment costs, which pose barriers to large-scale implementation.

Beyond Radio Frequency (RF)-based solutions, VLC has emerged as a promising complementary technology. Although VLC has been applied in vehicular communication and positioning, its use in adaptive traffic signal control remains limited. In this direction, Vehicular VLC (V-VLC) systems have been proposed, combining mesh networking and cellular infrastructures to improve message relaying and leverage edge computing. Some VLC applications and potential for vehicular networks will be discussed in detail in Section 2.5.

Connected Vehicle (CV) technologies are transforming urban mobility by enabling continuous, real-time communication between vehicles and infrastructure, which improves traffic efficiency, alleviates congestion, and increases safety. Recent solutions such as ATSC [29], Connected and Autonomous Vehicles (CAVs) [30], and reinforcement learning (RL)-based control approaches illustrate this shift toward intelligent traffic management. ATSC systems rely on sensor data to optimize signal timings, while CAVs can support advanced functions such as speed harmonization and cooperative maneuvers to minimize queues and delays. Nevertheless, these systems face challenges regarding interoperability, infrastructure costs, and their scalability to complex metropolitan networks.

Progress in wireless communication and V2V/V2I technologies opens opportunities to integrate traffic signal control with driving behaviors, enabling more comprehensive management of urban mobility [31]. Building on this perspective, this work introduces a framework that combines VLC-based localization with RL-driven traffic signal control.

The objective is to improve both vehicle and pedestrian flows across multiple intersections, reducing waiting times while increasing overall safety [32]. The proposed V-VLC system is assessed through agent-based simulations using Simulation of Urban MObility (SUMO) platform [33].

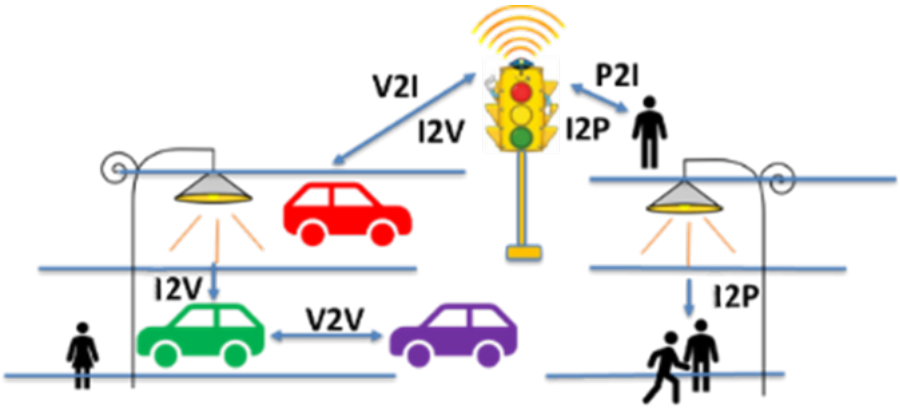


Figure 2.2: *2D representation of a traffic environment with VLC communication*

2.5 Visible Light Communication

As mentioned earlier, the main goal of ITS is to improve road safety, traffic efficiency, and driving comfort. By relying on I2V/V2I and V2V communications, ITS continuously collects, processes, and shares traffic information to enhance vehicle awareness. This flow of information supports a more effective management of transportation networks, improving efficiency and reducing traffic congestion [34]. The collected data can then be used to

dynamically adapt the operation of the system to changing traffic conditions [35, 36].

For ITS to be effective, it must be widely deployed, with smart vehicles and infrastructures spread across large areas to collect and share data efficiently. Although RF communications can support this exchange, the rapid growth of mobile data traffic in recent decades has revealed the limits of relying only on RF. Even with advanced reuse techniques, the available spectrum is insufficient to meet demand. In contrast, the visible light spectrum offers vast, unlicensed bandwidth that remains largely unused. VLC can therefore complement RF systems in building high-capacity communication networks. Because of this, VLC has recently gained significant attention as a complementary communication technology. By reusing existing LED-based devices such as traffic lights, street lamps, and vehicle headlights, VLC enables both illumination and high-speed data transmission simultaneously [37, 38]. This technology offers several inherent benefits, including wide bandwidth, low latency, enhanced security, and cost efficiency.

When integrated with V2V and V2I communication systems, VLC can support safety applications that, according to reports, could potentially prevent up to 81% of light-vehicle target crashes [39]. Beyond this, VLC has the capacity to strengthen vehicular networks under high-density traffic, improving road safety while enabling the efficient dissemination of critical traffic information to intelligent vehicles.

VLC can have indoor and outdoor applications. The first one has received more attention due to the great evolution of the concept of Light Fidelity (Li-Fi), which is a bidirectional wireless system that transmits data via visible light or IR. The second application, and the one that will be studied in this project, has been developed more slowly as it faces more challenges in terms of the environment in which it is used, the type of mobility and the weather conditions. Among outdoor scenarios, ITS represents the most relevant application of VLC, as vehicular networks can benefit directly from LED-based communication to implement Vehicular Visible Light Communication (V-VLC). In I2V/V2I applications, the focus lies on using traffic related infrastructure, such as streetlights and traffic signals, to deliver useful information.

Traffic signals are always active, which makes them especially suitable for applications related to safety and the broadcasting of traffic information. In V2V scenarios, VLC typically relies on the headlights and taillights of vehicles as transmitters, while photodiodes or onboard cameras act as receivers, ensuring direct and reliable communication between vehicles.

For pedestrian interaction, the architecture Pedestrian-to-Infrastructure (P2I) and Infrastructure-to-Pedestrian (I2P) communication. Pedestrians transmit crossing requests through VLC-enabled devices, and the infrastructure responds with trajectory assignments and safe crossing phase allocations.

Figure 2.3 shows a schematic representation of an intelligent traffic system supported by VLC communication. The illustration includes V2V, V2I and I2V interactions using LED-equipped streetlights and traffic signals. Additionally, P2I and I2P communication are represented, where pedestrians transmit crossing requests through VLC-enabled devices

and receive safe crossing phase allocations in return.

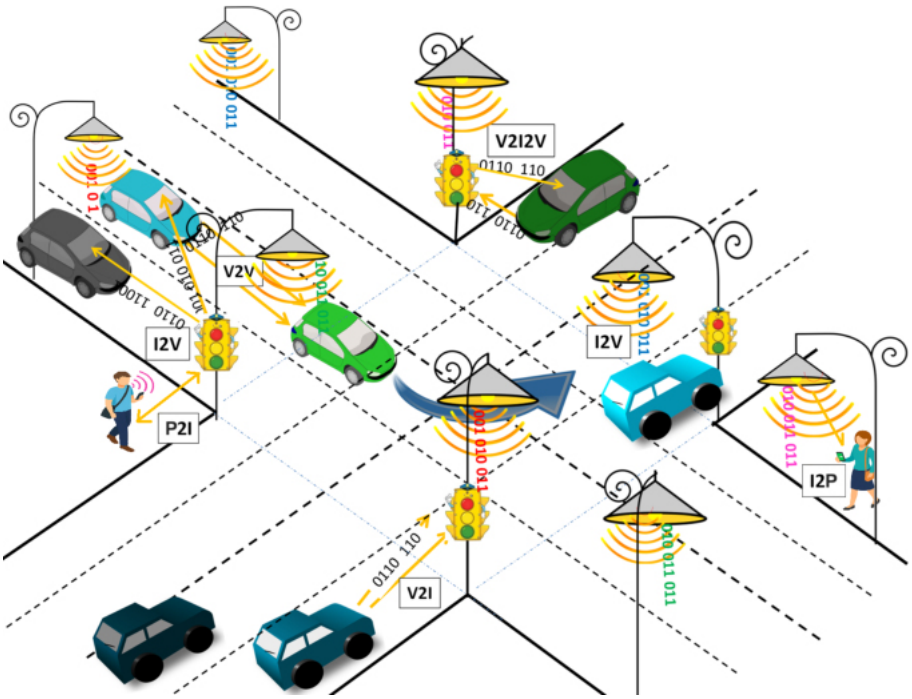


Figure 2.3: *Representation of an intelligent urban traffic system with VLC communication*

3 Simulation Environment and Traffic Scenarios

3.1 Simulation of Urban MObility (SUMO)

To evaluate the effectiveness of the proposed MARL methods and V-VLC system, agent-based simulations were conducted using the Simulation of Urban MObility (SUMO), an open-source, microscopic, and multimodal traffic simulator widely used in both academia and industry. SUMO provides a flexible and extensible platform to reproduce realistic traffic scenarios, including vehicles, pedestrians, and traffic signals, while offering fine-grained control of simulation parameters. Its open-source nature and active community make it particularly suitable for research in Intelligent Transportation Systems (ITS), where repeatability and customization are essential.

The choice of SUMO for this work is motivated by its strong integration with external control frameworks, especially through Traffic Control Interface (TraCI). TraCI allows external programs to interact with the simulator in real time by retrieving states of the environment and applying actions to the traffic lights. This feature makes SUMO particularly well-suited for the implementation and evaluation of adaptive and intelligent traffic signal control systems.

In this study, a traffic network was modeled. The network was generated using SUMO's network generation tools, where road segments, lanes, and intersection layouts were defined to emulate a small urban area. Traffic demand was created by specifying different route scenarios and vehicular flow to reflect different conditions such as peak and off-peak hours. This ensures that the environment provides heterogeneous and dynamic traffic patterns, which are necessary to test the adaptability of reinforcement learning algorithms.

Through the TraCI API, reinforcement learning agents interact with the simulator in real time by observing traffic states, such as queue lengths, vehicle speeds, and waiting times, and by applying signal control actions. During each simulation step, system states, actions, and rewards are exchanged, and performance metrics including vehicle and pedestrian waiting time, queue length, throughput, and speed are collected, providing a comprehensive view of how the system responds to different control strategies and serving as the basis for comparing the proposed reinforcement learning approaches.

3.2 Single Intersection Model

To illustrate the operation of the proposed framework, a single four-arm intersection was modeled in SUMO, serving as the fundamental building block of the larger traffic network. Each approach to the junction consists of two incoming and two outgoing lanes, allowing vehicles to execute right-turn, through, and left-turn movements. The rightmost lane enables vehicles to turn right or continue straight, while the leftmost lane is exclusively reserved for left turns. Vehicles must position themselves in the correct lane before reaching the stop line to ensure proper maneuver execution. This configuration reflects a typical arterial layout while being optimized for CAVs, where communication and precise control are assumed.

Figure 3.1 shows a screenshot of the modeled intersection in the SUMO simulator.

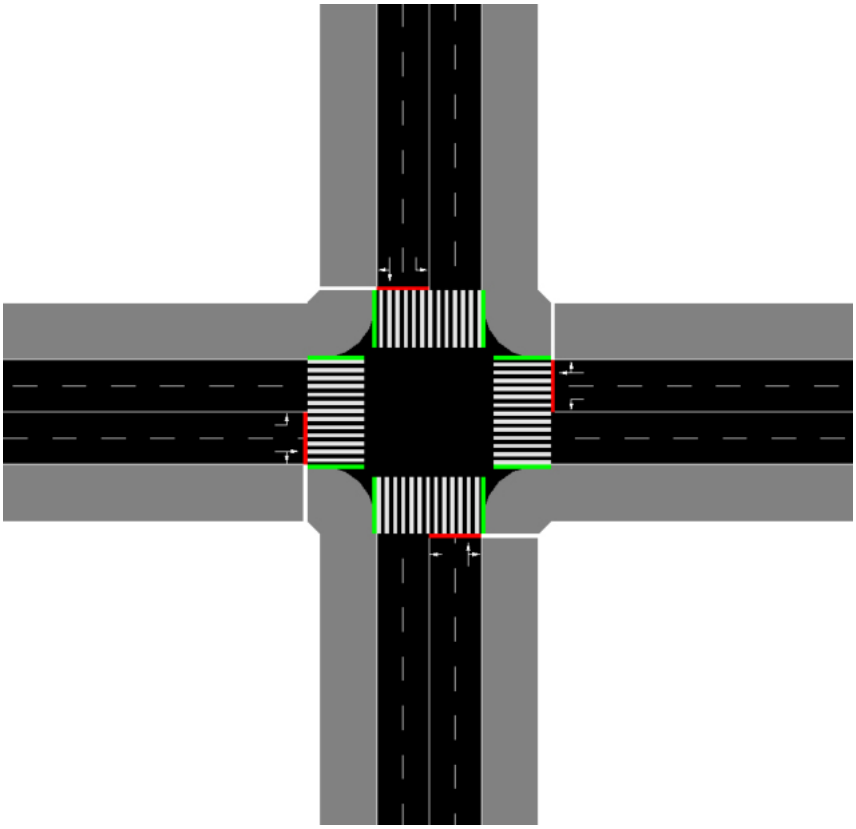


Figure 3.1: *Screenshot of the intersection modeled in SUMO.*

In the center of the intersection, a traffic light system controlled by the Intersection Manager (IM) regulates the movement of vehicles and pedestrians. A schematic representation of the intersection layout, including lane identifiers (L/0–7) and traffic light signals (TL/0–15), is provided in Figure 3.2.

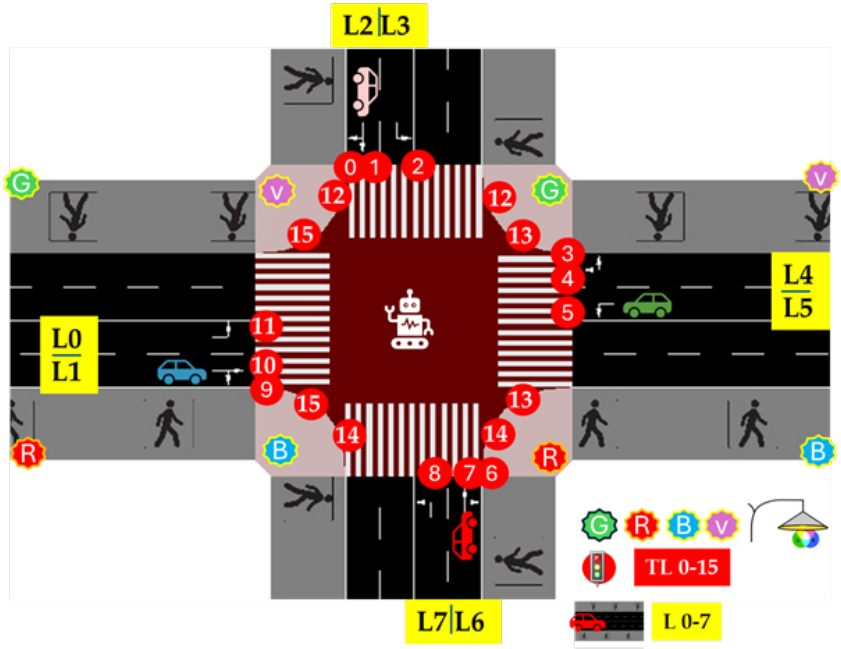


Figure 3.2: *Schematic diagram of a four-arm intersection with coded lanes (L/0–7) and traffic lights (TL/0–15).*

3.2.1 Traffic Signal Phases

Traffic control at the intersection is organized into nine possible phases, each corresponding to a set of traffic lights (TL) that turn green simultaneously, while all conflicting signals remain red. Unlike conventional fixed-order systems, phases are dynamically selected based on current traffic demand. Figure 3.3 illustrates the set of available phases.

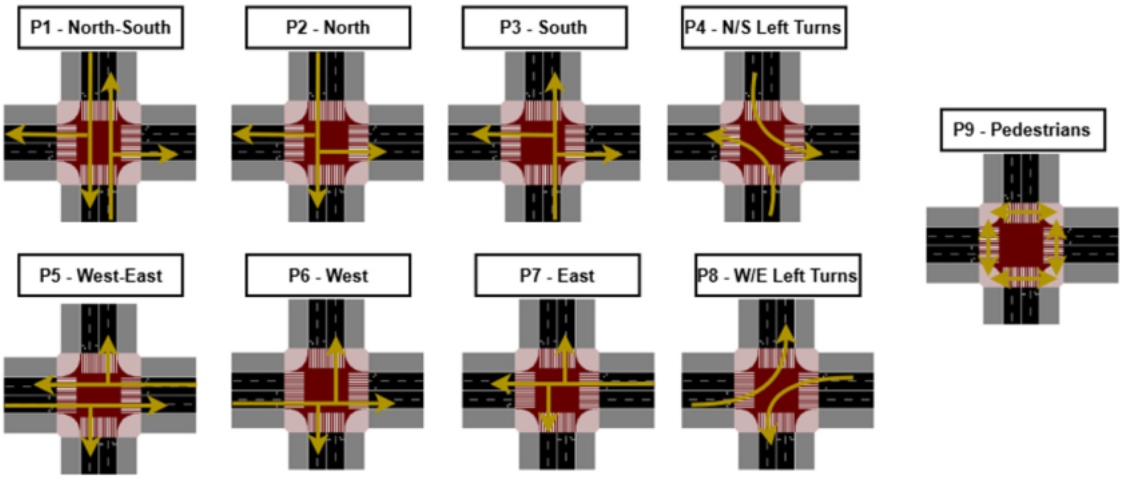


Figure 3.3: *Traffic signal phases for the modeled intersection, including vehicle and pedestrian movements.*

For the North–South alignment, Phase 1 activates TL/0–1 and TL/6–7 for straight and right-turn movements, while Phase 2 (TL/0–2) enables all northbound directions and Phase 3 (TL/6–8) enables all southbound directions. Left turns for this arterial are handled in Phase 4, with TL/2 and TL/8 set to green.

The West–East axis is managed through similar configurations. Phase 5 activates TL/3–4 and TL/9–10 for straight and right-turn movements, Phase 6 (TL/3–5) enables all east-bound directions, and Phase 7 (TL/9–11) enables all westbound directions. Phase 8 again provides straight and right-turn movements for E–W traffic (TL/3–4 and TL/9–10).

Finally, Phase 9 corresponds to pedestrian movement, with TL/12–15 set to green while all vehicle lights remain red. This guarantees full pedestrian priority and avoids overlaps between pedestrian and vehicular flows.

Table 3.1 summarizes the green signal allocation for each phase.

Table 3.1: Traffic light configuration for each phase.

Phase	**Name**	**Active Traffic Lights**
Ph 1	North–South straight/right	TL/0, TL/1, TL/6, TL/7
Ph 2	North to all directions	TL/0, TL/1, TL/2
Ph 3	South to all directions	TL/6, TL/7, TL/8
Ph 4	North–South left turns	TL/2, TL/8
Ph 5	West–East straight/right	TL/3, TL/4, TL/9, TL/10
Ph 6	East to all directions	TL/3, TL/4, TL/5
Ph 7	West to all directions	TL/9, TL/10, TL/11
Ph 8	West–East straight/right	TL/3, TL/4, TL/9, TL/10
Ph 9	Pedestrian crossing	TL/12, TL/13, TL/14, TL/15

Phase Timing: Each phase follows a fixed duration: green lights are active for 8 seconds, followed by a 4-second yellow interval. When the IM selects the same phase that was active in the previous cycle, the yellow interval is skipped and the green light continues without interruption. This prevents unnecessary stops while still ensuring safe transitions.

3.2.2 State Representation

The state representation adopted in this work architecture encodes the traffic environment by combining three complementary sources of information: vehicle position, vehicle speed, and pedestrian activity.

The road network approaching the intersection is discretized into a grid of 80 positional cells, obtained by dividing each of the eight incoming lane groups into ten segments, that indicate where vehicles are located as they approach the stop line. Each vehicle detected is assigned to one of these cells, allowing the system to capture queue formation and spatial distribution. Parallel to this, another 80 cells are maintained to store velocity information, where the speed of vehicles in each segment is normalized with respect to the maximum legal speed and averaged into the corresponding cell. In this way, it allows the model to distinguish between free-flowing traffic and queues. To complement these features, four additional inputs are dedicated to pedestrian demand, one for each crosswalk, which are activated when pedestrians are detected waiting to cross.

Together, the resulting state vector has 164 features - 80 for vehicle positions, 80 for vehicle speeds, and 4 for pedestrians. This representation captures a compact description of the traffic environment and serves as input to the IM.

3.3 Network Intersection Model

The experimental environment extends beyond a single intersection to a simplified urban traffic network composed of five interconnected four-arm junctions. Two main arterial roads are considered: a horizontal corridor (C0–C1–C2) and a vertical corridor (C3–C1–C4), which intersect at the central junction C1. The schematic of the traffic scenario used in this study is shown in Figure 3.4.

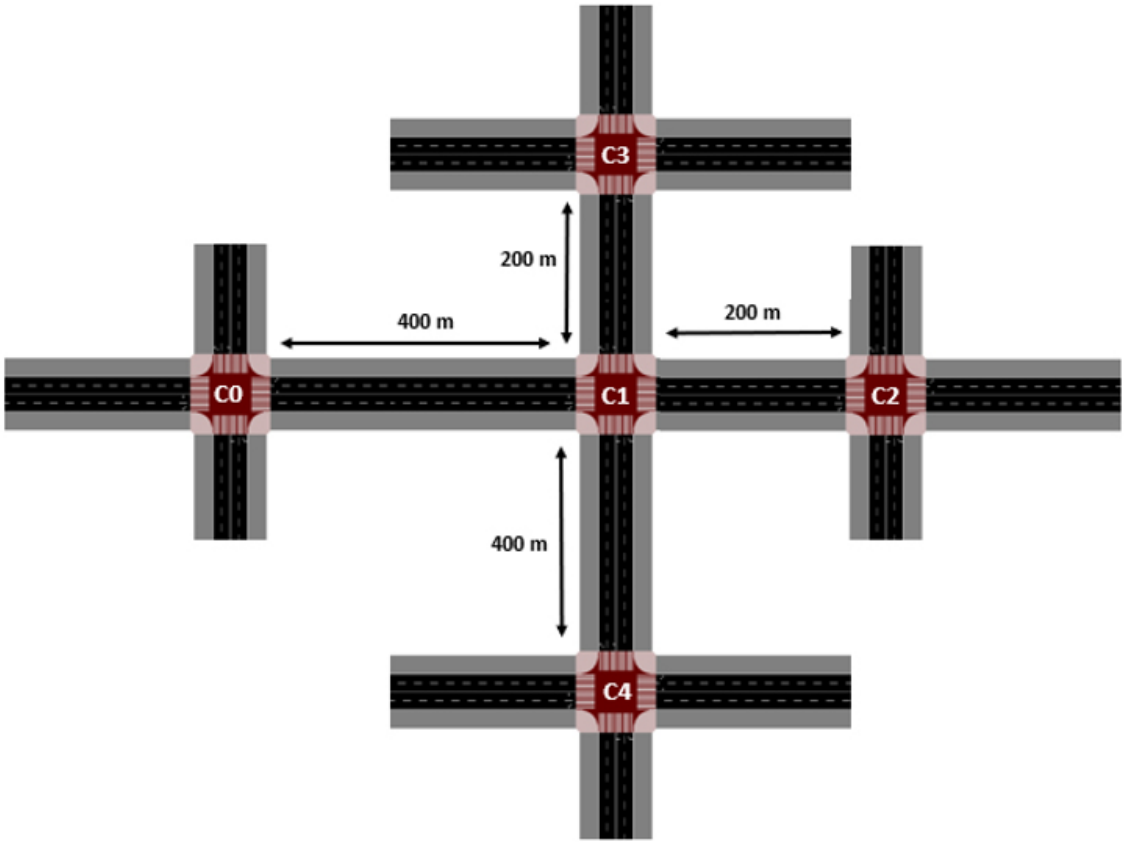


Figure 3.4: *Traffic scenario consisting of 5 homogeneous intersections with 4 arms each.*

The spacing between intersections was designed to reproduce realistic urban conditions. The distance between C1 and C0, as well as between C1 and C4, is 400 meters, while the segments connecting C1 to C2 and C3 measure 200 meters. These varying distances create heterogeneous traffic conditions across the network, influencing queue formation and travel times depending on the corridor and demand levels.

A defining feature of this setup is the role of intersection C1. Unlike the surrounding nodes, C1 does not introduce any local inflows of traffic. Instead, it functions purely as a transfer hub, receiving vehicles and pedestrians from the four adjacent intersections. All flows into its lanes are therefore determined by the signal decisions of neighboring controllers (C0, C2, C3, and C4). As a result, C1 becomes a pivotal coordination point: its activation choices mediate the balance between the two arterial axes, directly affecting congestion, throughput, and fairness across the network. Proper alignment at C1 can distribute flows evenly and prevent bottlenecks, while miscoordination may amplify imbalances and degrade

overall system performance.

To manage this environment, each of the five intersections is controlled by a dedicated MARL agent. Through VLC-based communication, agents continuously monitor local conditions — including vehicle positions, speeds, and pedestrian demand — and adapt their signal phases accordingly. This distributed learning architecture enables both local optimization and implicit cooperation, with C1 acting as the key hub whose policies can determine whether the system reaches a balanced equilibrium or falls into congestion.

Taking this configuration into account, the entire network state can be encoded as a 164 $\times$ 5 dimensional vector, corresponding to the concatenation of the local 164-dimensional state representation of each of the five intersections, resulting in a total of 820 features. In other words, with this representation it is possible to capture the complete environment, creating a description that will serve as input to the neural networks used in the studied algorithms, which will be detailed in Chapter 4 (Implemented Algorithms).

Vehicle Dynamics: All vehicles in the simulation are modeled as passenger cars following SUMO's default car-following and lane-changing models. A maximum speed of 13.89 m/s (50 km/h) was defined for all lanes, consistent with typical urban road limits. Acceleration and deceleration parameters follow SUMO's default configuration, ensuring realistic driving dynamics. Pedestrian agents were assigned a nominal walking speed of 1.0 m/s (3.6 km/h), representing average urban walking behavior.

3.3.1 Traffic Scenarios

In order to evaluate the behavior of the proposed system under different operating conditions, three traffic scenarios were designed. Each episode simulates one hour of traffic, corresponding to 3600 seconds of simulation time. Within this period, a fixed flow of 2000 pedestrians is introduced, while the number of vehicles varies according to the scenario. The low-demand case generates 1800 vehicles, representing light traffic conditions with limited congestion. The medium-demand case increases the load to 2200 vehicles, capturing more balanced urban conditions. Finally, the high-demand case reaches 2600 vehicles, providing a stress test for the decision-making capabilities of the system.

These values are defined per simulation episode, ensuring controlled and comparable experiments, enabling a systematic analysis of how the agents learn and adapt their policies to different traffic intensities.

A summary of the traffic demand configurations for each scenario is provided in Table 3.2.

Table 3.2: Traffic scenarios used in the experiments.

Scenario	**Number of Vehicles**	**Number of Pedestrians**
Low	1800	2000
Medium	2200	2000
High	2600	2000

3.3.2 Traffic Control Strategies

To evaluate the adaptability of the proposed MARL framework under different traffic conditions, three traffic control strategies were defined, each represented by a distinct traffic generation pattern. The strategies differ in how vehicles are distributed between the circular road (W–E: C0–C1–C2) and the radial road (N–S: C3–C1–C4), as well as in the proportion of northbound and southbound flows along the radial axis.

Vehicle generation follows a stochastic process, where most vehicles (75%) are assigned straight-through routes while the remaining 25% are allocated to turning maneuvers. The directional bias of each strategy is implemented by adjusting the relative probability of sampling circular versus radial routes, and, for the radial flows, the probability of choosing northbound or southbound directions.

The three strategies are summarized as follows:

- **Strategy 1 – Balanced Distribution:** Traffic generation is distributed evenly across the network. Circular (W–E) and radial (N–S) arteries are equally likely to be selected, and radial flows are balanced between inbound and outbound directions. This scenario serves as the baseline for fairness and throughput.
- **Strategy 2 – Radial + Northbound Distribution:** The overall generation rate is higher on the radial artery (65%) than on the circular artery (35%). Among the radial vehicles, 75% are generated at C4 (south–north movement) and 25% at C3.
- **Strategy 3 – Radial + Southbound Distribution:** Follows the same generation proportions as Strategy 2 (65% radial, 35% circular), but with the radial flow reversed: 75% of vehicles are generated in the north–south direction at intersection C3, and 25% at C4.

4 Implemented Algorithms

4.1 Deep Q-Learning

Deep Q-Learning (DQN) is a value-based reinforcement learning algorithm that extends the classical Q-Learning by using a deep neural network to approximate the action-value function $Q(s, a)$. Instead of storing values for each state-action pair in a table, which becomes infeasible in high-dimensional environments, DQN learns a function $Q_\theta(s, a)$ parameterized by the network weights θ.

At each time step t, the agent observes a state s_t, selects an action a_t according to an ϵ-greedy policy, receives a reward r_t, and transitions to the next state s_{t+1}. The experience tuple (s_t, a_t, r_t, s_{t+1}) is stored in a *replay buffer*. During training, random minibatches are sampled from this buffer to break temporal correlations between consecutive experiences and stabilize learning. To further improve stability, DQN maintains a *target network* Q_{θ^-}, which is updated periodically by copying the weights of the online network Q_θ.

Figure 4.1 illustrates the architecture of the IM, which is composed of a decentralized neural network trained based on the observations and experiences of individual agents. Each agent is responsible for controlling its own intersection. This enables real-time decision-making, dynamically adjusting the active signal phases according to the observed traffic flows on each approach, thereby optimizing traffic movement within the cell.

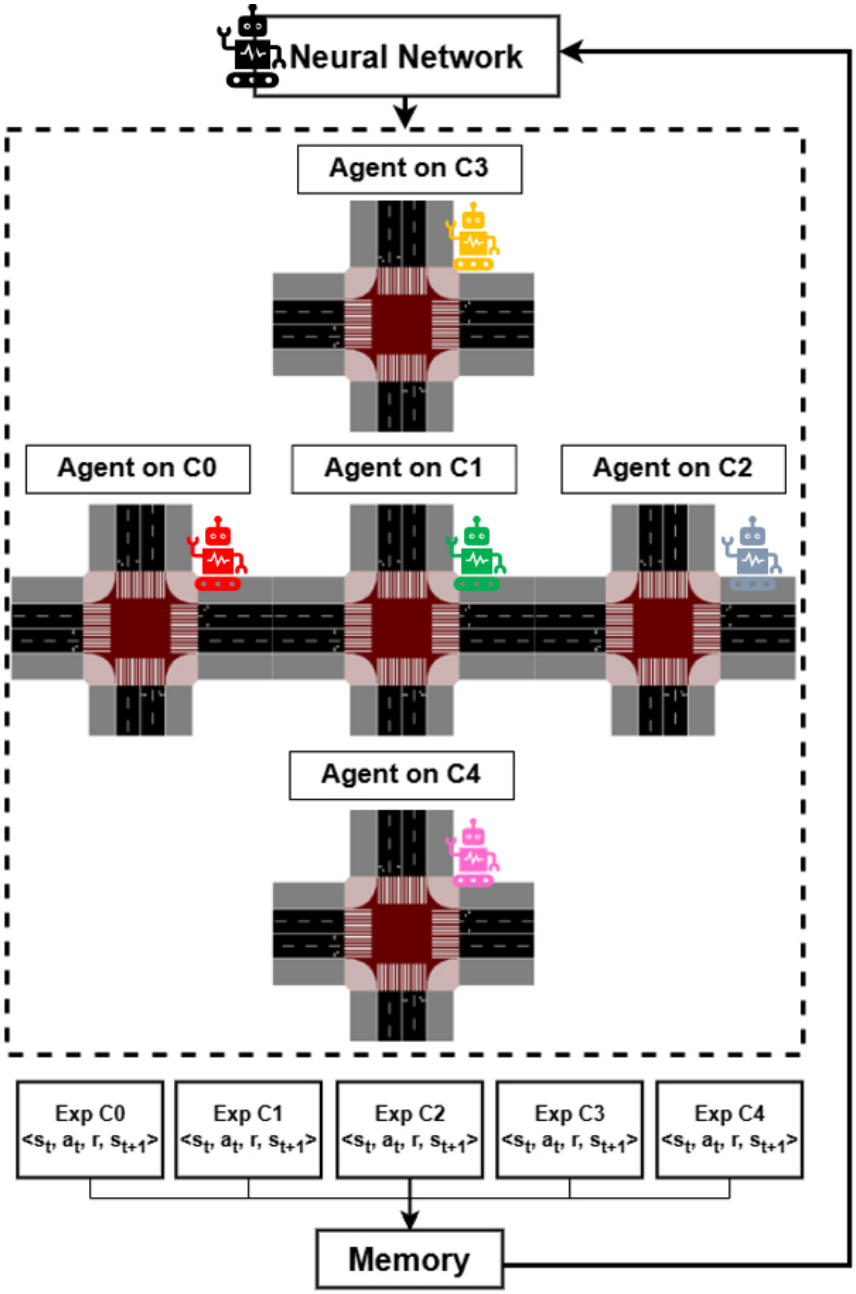


Figure 4.1: *Intersection Manager architecture based on a Deep Neural Network.*

Unlike on-policy methods, which require new trajectories to update the policy, DQN is an *off-policy* algorithm. This means that the agent can learn from past experiences generated by an older policy or even from exploratory actions that do not follow the current greedy policy. This property improves data efficiency but also makes training more challenging in multi-agent environments, where the behavior of other agents changes over time and leads to non-stationarity.

The update of the Q-values in DQN follows the Bellman equation. The target value for a given state-action pair is defined as:

$$Q_{\text{target}}(s_t, a_t) = r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a'), \tag{4.1}$$

where r_t is the reward at time t, γ is the discount factor, and s_{t+1} is the next state. The parameters θ^- correspond to the target network, which is a delayed copy of the online network used to generate more stable target values. The learning objective is to minimize the difference between the predicted Q-values $Q_\theta(s, a)$ and these target values, using a Mean Squared Error (MSE) loss function:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \Big(Q_{\text{target}} - Q_\theta(s, a) \Big)^2, \tag{4.2}$$

where N is the batch size, Q_{target} is the target Q-value computed using the Bellman equation, and $Q_\theta(s, a)$ is the predicted Q-value output by the neural network.

4.1.1 Application to Traffic Signal Control

In the traffic signal control context, each agent represents one intersection. The state representation described in Chapter 3.2.2 is used as the input to the neural network. This includes spatial features such as vehicle positions, velocities, and pedestrian activity, which together provide a high-dimensional description of traffic conditions at the intersection.

The output layer of the DQN corresponds to the set of nine possible actions available to the agent. In this approach, actions are defined as the selection of traffic signal phases, i.e., deciding which phase to activate at the next time step. Each output neuron therefore represents the Q-value associated with choosing a specific phase, and the agent selects the action that maximizes this value, subject to an ϵ-greedy exploration strategy.

The reward function is designed to promote efficient traffic flow while ensuring pedestrian safety. At each time step t, the reward is defined as the weighted decrease in accumulated waiting times of both vehicles and pedestrians:

$$r_t = p_{\text{veh}} \cdot \Big(ATWT_{\text{veh},t-1} - ATWT_{\text{veh},t} \Big) + p_{\text{ped}} \cdot \Big(ATWT_{\text{ped},t-1} - ATWT_{\text{ped},t} \Big), \tag{4.3}$$

where p_{veh} and p_{ped} are weighting factors that balance the relative importance of vehicles and pedestrians. In this work, both values are set to $p_{\text{veh}} = p_{\text{ped}} = 0.5$, giving equal priority to vehicles and pedestrians. $ATWT_{\text{veh},t}$ and $ATWT_{\text{ped},t}$ denote the *accumulated waiting times* of all vehicles and pedestrians at time t, respectively.

These quantities are computed as:

$$ATWT_{\text{veh},t} = \sum_{i=1}^{n_{\text{veh}}} wt(\text{veh}_i, t), \quad ATWT_{\text{ped},t} = \sum_{j=1}^{n_{\text{ped}}} wt(\text{ped}_j, t), \tag{4.4}$$

where $wt(\cdot, t)$ represents the waiting time of an individual vehicle or pedestrian at step t, defined as the number of seconds during which the agent's speed remains below 0.1 m/s since its entry into the environment. n_{veh} and n_{ped} represent the total number of vehicles and pedestrians present in the network at time t.

This reward formulation encourages the agent to minimize overall waiting times across both modes of traffic. By including pedestrians explicitly, the model can achieve a fairer balance between vehicle throughput and pedestrian safety, avoiding situations where one group is systematically disadvantaged.

The standard DQN algorithm is outlined in Algorithm 1.

Algoritmo 1 Deep Q-Learning (DQN)

Require: Discount factor γ, exploration schedule ϵ, batch size N, target update period C

Initialize replay buffer $\mathcal{D}$
Initialize online network Q_θ with random weights
Initialize target network $Q_{\theta^-} \leftarrow Q_\theta$
for episode $= 1 \dots M$ **do**
    $s \leftarrow env.reset()$
    for $t = 1 \dots T$ **do**
        Select $a \leftarrow \epsilon$-greedy(Q_θ, s)
        Execute a, observe r and s'
        Store (s, a, r, s') in $\mathcal{D}$
        Sample minibatch of size N from $\mathcal{D}$
        Compute targets: $Q_{target} \leftarrow r + \gamma \max_{a'} Q_{\theta^-}(s', a')$
        Update Q_θ minimizing $L(\theta) = \frac{1}{N} \sum (Q_{target} - Q_\theta(s, a))^2$
        if step mod$C = 0$ **then**
            $Q_{\theta^-} \leftarrow Q_\theta$
        end if
        $s \leftarrow s'$
    end for
end for

4.1.2 Network Architecture and Training Parameters

Each intersection is managed by a dedicated MARL agent that perceives its local environment — collecting data on vehicles and pedestrians via VLC-based communication — and cooperates with neighboring agents through shared information. All agents share the same Deep Q-Network (DQN) network, which learns to select the optimal signal phase at each intersection based on the maximization of expected cumulative rewards. This parameter sharing strategy improves scalability and ensures that the learned policy can be generalized across the network.

The neural network receives as input the 164-dimensional state representation described earlier. These inputs feed into a stack of 2 hidden layers, each with 400 neurons and Rectified Linear Unit (ReLU) activations, which provide sufficient representational capacity to capture the complexity of traffic dynamics. The final layer is the output layer with nine neurons, each corresponding to one possible traffic signal phase. The agent selects its next action by choosing the phase associated with the maximum Q-value, subject to the ϵ-greedy exploration strategy.

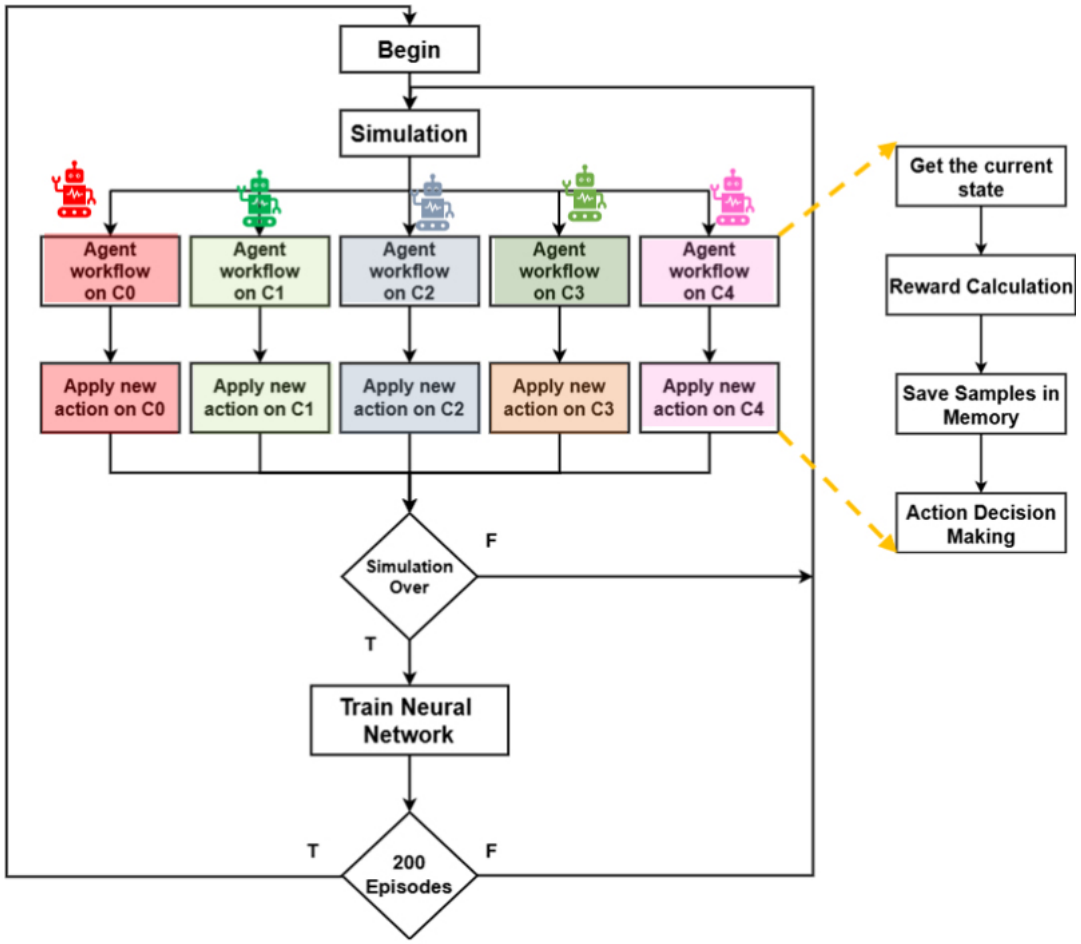


Figure 4.2: *Workflow of the proposed DQN-based multi-agent framework.*

The training of the DQN agents relies on several hyperparameters that directly influence performance and stability. The learning rate ($\alpha = 0.001$) controls the step size of weight updates in the neural network; this relatively small value was chosen to ensure stable convergence while avoiding oscillations during training. The discount factor ($\gamma = 0.75$) defines the importance of future rewards compared to immediate ones; in this context, a lower value emphasizes short-term reductions in waiting time, which is desirable for responsive traffic signal control. The exploration rate (ϵ) starts at 1.0 and gradually decreases towards 0 across training, ensuring sufficient exploration before convergence to stable policies. The replay buffer stores up to 50,000 transitions, allowing the algorithm to learn from a diverse set of past experiences and reducing correlations between consecutive samples. From this buffer, minibatches of size $N = 100$ are sampled for each update, providing a balance between stable gradient estimation and computational efficiency. The target network is updated every 20 training epochs, which helps stabilize learning by reducing oscillations in the target values. The model is trained for 200 episodes, each consisting of 3,600 steps, using the Adam optimizer with the MSE loss function penalizing deviations between predicted and target Q-values. The full set of parameters is summarized in Table 4.1.

Table 4.1: Hyperparameters used in Deep Q-Learning training.

Parameter	**Value**
Learning rate (α)	0.001
Exploration rate (ϵ)	$1.0 \rightarrow 0.00$ (linear decay)
Replay buffer size	50,000
Batch size (N)	100
Target network update frequency	Every 20 training epochs
Training epochs (per episode)	500
Training episodes	200
Max steps per episode	3,600
Neural network architecture	2 hidden layers, 400 neurons each, ReLU
Loss function	MSE
Optimizer	Adam

Discussion: By combining high-dimensional state representations with neural function approximation, DQN agents can learn effective traffic signal policies that reduce congestion and improve throughput. However, a key limitation in multi-agent deployments is that agents operate solely on local observations, without explicit knowledge of surrounding intersections or the states of neighboring agents. This lack of contextual information can restrict global coordination, as neighboring conditions often provide valuable indicators for optimizing traffic flow across the ambient.

To mitigate this, an extension was explored in which each agent integrates information from its direct neighbors into the learning process, improving coordination across adjacent intersections. A variation of this extension is discussed in detail in Section 4.2.

4.2 Deep Q-Learning with Q-value Transfer

In [19] it is proposed a Cooperative Deep Q-Learning Network with Q-value transfer (QT-CDQN), where the traffic scenario is modelled as a multi-agent reinforcement learning system. Each agent searches the optimal strategy to control an intersection by a deep Q-network, taking discrete state encoding of traffic information, like vehicle position in intersection approaching lanes and normalized speed as network inputs. To work cooperatively, agents consider the influence of the latest actions of its neighbors in the learning process. In particular, the predicted Q-values of the neighboring agents at the latest time step are transferred to the loss function.

In contrast to this formulation, this thesis proposes a slightly different variation. Instead of injecting the neighbors' Q-values into the loss function, they are incorporated directly into the target calculation used for the Bellman update. This is done by adding a term that aggregates the predicted Q-values from these neighbors. The Q-target values are calculated based on

$$Q_{\text{target}} = r_t + \gamma \cdot \max_{a'} \left[Q_{\theta^-}(s_{t+1}, a') + \beta \cdot \frac{1}{N} \sum_{n=1}^{N} Q_{\theta^-}(s_{t+1}^n, a') \right] \tag{4.5}$$

Where β is a weighting factor that regulates the influence of these neighbors on the Q-value update. This means that when $\beta = 0.0$, this equation is equivalent to Equation 4.1. The coefficient β regulates the influence of this cooperative component, while the normalization factor $\frac{1}{N}$ ensures that the contribution of the neighbors remains balanced regardless of their number. In this formulation, s^n_{t+1} denotes the next state observed by neighbor n at time step $t+1$, and $Q_{\theta^-}(s^n_{t+1}, a')$ represents the predicted Q-values of those neighbor states obtained through the target network. This guarantees that the cooperative term is consistent with the Bellman update and remains stable during training.

A fair value of β promotes cooperation that benefits not only the individual agent but also its neighbors, fostering a coordinated global traffic control strategy. In this work, β was set to 0.3, as other values were tested and resulted in poorer performance. Higher β values encourage more cooperative decisions, benefiting neighbors but potentially at the expense of the individual agent's own performance, while a lower β favors more independent, locally optimized actions.

The proposed approach is presented in Algorithm 2, which details the training procedure with the additional cooperative component.

Algoritmo 2 Deep Q-Learning with Q-value Transfer (QT-DQN)

Require: Discount factor γ, exploration schedule ϵ, batch size N, target update period C, cooperation weight β

Initialize replay buffer $\mathcal{D}$
Initialize online network Q_θ
Initialize target network $Q_{\theta^-} \leftarrow Q_\theta$
for episode $= 1 \dots M$ **do**
  $s \leftarrow env.reset()$
  for $t = 1 \dots T$ **do**
    Select $a \leftarrow \epsilon$-greedy(Q_θ, s)
    Execute a, observe r, s', and neighbors' states $\{s'^n\}$
    Normalize neighbor actions with `swap_actions`
    Store $(s, a, r, s', \{s'^n\})$ in $\mathcal{D}$
    Sample minibatch of size N from $\mathcal{D}$
    Compute cooperative targets:

$$Q_{\text{target}} \leftarrow r + \gamma \max_{a'} \left(Q_{\theta^-}(s', a') + \beta \cdot \frac{1}{|\mathcal{N}|} \sum_n Q_{\theta^-}(s'^n, a') \right)$$

    Update Q_θ minimizing $L(\theta) = \frac{1}{N} \sum (Q_{\text{target}} - Q_\theta(s, a))^2$
    if step $\bmod C = 0$ **then**
      $Q_{\theta^-} \leftarrow Q_\theta$
    end if
    $s \leftarrow s'$
  end for
end for

4.2.1 Differences Compared to Standard DQN

This design choice was made to preserve the scalability and simplicity of the DQN framework, allowing agents to benefit from additional context without requiring a dedicated

network per intersection or major architectural modifications.

In terms of training, the configuration largely follows the standard DQN architecture presented in Section 4.1, with the addition of the cooperative target formulation and the number of training episodes. This method is feasible due to the homogeneity of the intersections, allowing similar observations across agents to be leveraged collectively. Furthermore, by incorporating neighbor influence into the learning process, the approach aims to promote coordination between adjacent intersections, enhancing scalability and adaptability within the network.

In this work, agents were trained for 175 episodes instead of 200, as this proved to be a more efficient choice. Beyond this point, the cumulative negative reward occasionally exhibited oscillations, indicating that additional training did not consistently lead to performance improvements. On average, training with DQN took about 18 hours in the low-demand scenario, 20 hours in the medium-demand scenario, and 22 hours in the high-demand scenario. QT-DQN required approximately 16, 18, and 20 hours for the same scenarios. Both algorithms had a similar training time per episode, with the difference in total time explained by the reduced number of episodes in QT-DQN.

4.2.2 Practical Implications and Limitations

In practice, this formulation provides the agent with additional context that helps it distinguish when vehicles are being received from neighboring intersections and when they are not. With this cooperative signal incorporated into the training targets, the agent is expected to learn more informed policies that anticipate incoming flows rather than optimizing only for its own immediate conditions.

Nevertheless, some limitations remain. First, neighbors are only considered during training through the target update equation; the input to the neural network remains limited to the agent's own local state representation of 164 features. As a result, during testing and deployment, decisions are made solely on the local state, without direct access to neighbors' observations.

Second, since neighbor Q-values are simply summed and averaged, the association between each Q-value and its originating neighbor is lost, preventing the agent from identifying which neighbor produced a given value. This is problematic because the same action may not represent the same traffic dynamics depending on the orientation of the neighbor. For example, an *East–West* phase in a horizontally aligned neighbor implies that vehicles will enter the intersection under control, whereas in a vertically aligned neighbor it may represent the opposite. To address this, a normalization procedure was introduced through a function called `swap_actions` (available in Appendix A). This function reorders the action values of each neighbor so that all actions correspond uniformly across orientations before averaging. In this way, the aggregated Q-values are consistent and comparable, ensuring that cooperative information is meaningful and correctly aligned across the network.

4.3 MAPPO

Multi-Agent Proximal Policy Optimization (MAPPO) is an on-policy, policy-gradient based reinforcement learning algorithm designed for multi-agent environments. It extends the Proximal Policy Optimization (PPO) framework by introducing a centralized critic that conditions on the global state, while each agent maintains its own decentralized actor policy. This hybrid formulation combines the scalability of independent agents with the stability of centralized training, making MAPPO particularly suitable for cooperative tasks such as traffic signal control.

Unlike DQN, which is an off-policy algorithm that can learn from past experiences stored in a replay buffer, MAPPO is strictly on-policy. This means that policy updates are performed only with the most recent trajectories collected by the current policy, which improves stability and prevents the algorithm from learning outdated behaviors.

Formally, PPO optimizes the clipped surrogate objective:

$$L^{CLIP}(\theta) = \hat{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{4.6}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio between the updated and old policies, $\hat{A}_t$ is the estimated advantage at time t, and ϵ is a clipping parameter that limits the size of updates. The operator $\hat{E}_t[\cdot]$ denotes the empirical expectation over the collected batch of timesteps, i.e., the average across sampled trajectories. The clipping mechanism prevents excessively large policy updates, stabilizing training.

The advantage function is estimated using *Generalized Advantage Estimation* (GAE), which reduces variance while maintaining a low bias. Given the temporal-difference error

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t), \tag{4.7}$$

the advantage is computed as

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \tag{4.8}$$

where γ is the discount factor and λ controls the trade-off between bias and variance. Intuitively, a positive advantage indicates that the chosen action led to a better outcome than expected by the critic, while a negative advantage suggests the opposite. This signal is then used in the clipped objective to adjust the actor's policy in favor of more advantageous actions.

Loss functions: In MAPPO, the optimization process separates into two complementary objectives. The **actor loss** is based on the clipped surrogate function:

$$L^{\pi}(\theta) = -\hat{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) + c_{\text{ent}} \cdot \mathcal{H}[\pi_\theta(\cdot|s_t)] \right],$$

where the entropy term $\mathcal{H}$, weighted by c_{ent}, encourages exploration. The **critic loss** is a mean squared error between the predicted values and the estimated returns:

$$L^V(\phi) = c_{\text{vf}} \cdot \hat{E}_t\big[(R_t - V_\phi(s_t))^2\big].$$

Thus, while DQN and QT-DQN rely solely on an MSE loss to approximate Q-values, MAPPO combines a policy optimization loss for the actor with an MSE loss for the critic.

One-Hot Encoding (OHE) for Agent Identity: A key addition in this work is the use of OHE to represent each agent's identity as a binary vector where a single element is set to 1, uniquely identifying the agent, while all others are set to 0. By concatenating this vector to the actor's observation, the shared network can distinguish agents and specialize its policy to the specific characteristics. This is particularly important in heterogeneous traffic networks, where intersections may differ in geometry or traffic flow patterns. OHE allows a single shared actor to learn differentiated strategies without interference across agents.

In the DQN baseline, OHE was **not** included. Since all agents shared the same Q-network without any explicit identity features, the network was forced to approximate multiple distinct policies with a single parameter set. This sometimes led to policy aliasing, where similar local observations required different actions depending on the intersection, but the network could not disambiguate them. In principle, OHE could also have been added to DQN; however, its off-policy replay buffer would still mix non-stationary experiences from all agents, yielding unstable training. In this setting, adding OHE would merely increase input dimensionality without resolving the core instability. By contrast, MAPPO trains on-policy and leverages a centralized critic, which provides a more stable learning signal.

The MAPPO training procedure is summarized in Algorithm 3.

Algoritmo 3 Multi-Agent Proximal Policy Optimization (MAPPO)

Require: Discount factor γ, GAE parameter λ, clipping parameter ϵ, entropy coefficient c_{ent}, value loss coefficient c_{vf}, rollout horizon H, PPO epochs K, minibatch size M

Initialize shared actor network π_θ and centralized critic V_ϕ
for episode $= 1 \dots N_{\text{episodes}}$ **do**
    Collect rollouts of horizon H from all agents:
    for each step $t = 1 \dots H$ and agent i **do**
        Observe local state s_t^i and one-hot agent id
        Sample action $a_t^i \sim \pi_\theta(\cdot|s_t^i, id^i)$
        Execute a_t^i, observe reward r_t^i and next state s_{t+1}^i
    end for
    Compute advantages $\hat{A}_t$ using GAE with centralized critic V_ϕ
    Compute returns R_t for critic updates
    for epoch $= 1 \dots K$ **do**
        Sample minibatches of size M
        Update actor by minimizing clipped surrogate loss $L^{CLIP}(\theta)$
        Update critic by minimizing value loss $L^V(\phi)$
    end for
end for

4.3.1 Differences Compared to DQN

While DQN is a value-based method that approximates action-values $Q(s, a)$ through a neural network, MAPPO follows an actor–critic paradigm. The main differences and advantages can be summarized as follows:

- **On-policy training:** MAPPO discards old trajectories and updates policies only with the most recent rollouts, improving stability at the cost of lower sample efficiency.
- **Centralized Critic:** During training, the critic has access to the joint global state and can evaluate the combined actions of all agents. This enables MAPPO to capture interdependencies and promote better coordination across intersections.
- **Decentralized Actors:** Each agent's policy (actor) depends only on its own local observation, ensuring scalability and decentralized execution during deployment.
- **Stability:** The clipped surrogate objective prevents large, destabilizing policy updates, reducing oscillations often observed in value-based methods such as DQN.
- **Agent Identity Encoding:** In this implementation, MAPPO augments the shared actor input to 169 features with an explicit agent identity using OHE. This allows the network to specialize its policy for each intersection while still sharing parameters across agents.

4.3.2 Network Architecture and Training Parameters

The MAPPO framework employs a centralized critic and decentralized actors. The critic network takes the global observation of dimension `global_input_dim` (820) as input and consists of two fully connected hidden layers with 256 neurons each and ReLU activation. The output layer is a single linear unit that produces the scalar value function $V(s)$. The actor network receives the local observation of the agent concatenated with a one-hot encoding of the agent identifier, i.e., [`obs_local`, `one_hot(agent_id)`]. The actor uses two fully connected hidden layers with 128 neurons each and ReLU activation, followed by an output layer with `n_actions` (9) neurons producing raw logits. These logits are transformed through a softmax function to obtain a probability distribution over the discrete action space. Consequently, the critic outputs a scalar state-value estimate, while the actor outputs a vector of action probabilities.

The full details of the network design and training hyperparameters are summarized in Table 4.2.

Table 4.2: Actor–Critic network architectures for MAPPO.

Component	**Architecture / Details**
Actor input	obs_local $\in R^{\text{obs\_dim}}$ concatenated with one-hot agent
Actor hidden layers	2 fully-connected layers, width = 128, activation ReLU
Actor output	Logits over n_{actions} (categorical policy)
Action sampling	Stochastic (categorical); greedy for deterministic eval
Central Critic input	Global state $\in R^{\text{global\_input\_dim}}$
Central Critic hidden layers	2 fully-connected layers, width = 256, activation ReLU
Central Critic output	Scalar value $V(s)$
Value loss	MSE scaled by c_{vf}

The training of the MAPPO agents relies on several hyperparameters that balance stability, exploration, and computational efficiency. The discount factor ($\gamma = 0.75$) and GAE parameter ($\lambda = 0.75$) control how future rewards are valued and how the advantage estimates trade off variance against bias. The learning rates for both the actor and critic were set to 0.002, with the Adam optimizer applied to update the networks. Training proceeds in rollouts of horizon $H = 4096$ steps, after which policy and value updates are performed. Each rollout is reused for four PPO epochs, with a minibatch size equal to the horizon. The clipping parameter $\epsilon = 0.2$ stabilizes policy updates by limiting large changes in the probability ratio. Exploration is encouraged through an entropy term weighted by $c_{ent} = 0.01$, while the critic loss is scaled by $c_{vf} = 0.5$ to balance its influence. To prevent gradient explosion, updates are clipped to a maximum global norm of 0.5. Advantages are standardized before use, and invalid actions are masked in the logits by assigning large negative values.

The complete set of parameters is summarized in Table 4.3.

Table 4.3: Hyperparameters used in MAPPO training.

Parameter	**Value**
Total episodes	100
Max steps per episode	3,600
Discount factor (γ)	0.75
GAE parameter (λ)	0.75
Actor learning rate	0.002
Critic learning rate	0.002
Optimizer	Adam
Rollout horizon (steps per update)	4,096
PPO epochs per update	4
Minibatch size	4,096
Clipping parameter (ϵ)	0.2
Entropy coefficient (c_{ent})	0.01
Value loss coefficient (c_{vf})	0.5
Gradient clipping (global norm)	0.5

4.3.3 Application Considerations

In practice, MAPPO provides several advantages over DQN. The centralized critic can evaluate combinations of actions across intersections, improving coordination in situations where local decisions are interdependent. This helps avoid scenarios where one agent improves its local intersection at the expense of creating congestion in a neighboring one.

In terms of learning efficiency, MAPPO demonstrated a much faster convergence compared to both DQN and QT-DQN. While the value-based methods required a significantly larger number of episodes to stabilize, MAPPO consistently reached competitive performance within approximately 100 training episodes. This accelerated convergence greatly reduced the training requirements: on average, training a MAPPO model took around 10 hours, which is roughly half the time needed for training the DQN-based models.

By introducing the one-hot agent encoding through OHE, the actors can adapt their strategies to the specific geometry and traffic patterns of their intersections, rather than relying on a single shared representation. This additional flexibility enhances overall policy quality in heterogeneous or asymmetric networks.

In conclusion, MAPPO provides a more stable and coordinated learning framework compared to DQN, particularly in multi-agent traffic control settings.

5 Results and Discussion

This chapter presents the experimental evaluation of the three algorithms studied in this work: Deep Q-Learning (DQN), Deep Q-Learning with Q-value Transfer (QT-DQN), and Multi-Agent Proximal Policy Optimization (MAPPO). The evaluation is performed under the three traffic strategies introduced in Subsection 3.3.2 and across the three demand scenarios described in Subsection 3.3.1.

To assess and compare the performance of the algorithms, several traffic-related metrics are analyzed:

- **Pedestrian Curve**: number of pedestrians present in the environment over time.
- **Vehicle Curve**: number of vehicles present in the environment over time.
- **Halting Vehicles (C0–C4)**: number of vehicles halted at each intersection (a vehicle is considered halted if its speed is lower than 0.1 m/s).
- **Halting Pedestrians (C0–C4)**: number of pedestrians halted at each intersection (a pedestrian is considered halted if its speed is lower than 0.2 m/s).
- **Average Speed (C0–C4)**: average vehicle speed measured at each of the five intersections.
- **Environment Average Speed**: average of the five intersection speeds, providing an overall indicator of mobility within the environment.
- **Average Phase Distribution**: proportion of time each phase remains active at each intersection.

These metrics provide a comprehensive view of both vehicular and pedestrian traffic dynamics, capturing not only flow and throughput but also congestion levels, fairness, and phase allocation across the environment.

A total of 27 scenarios were evaluated, resulting from the combination of the three algorithms, three traffic strategies, and three demand levels. For each configuration, five independent simulation runs were performed, and the results were averaged.

In this chapter, the results are organized by traffic demand scenario. Each demand level (low, mid, and high) is analyzed separately, and within each scenario, the outcomes for the three control strategies are presented, covering all relevant performance metrics.

Among these, the mid-demand scenario (Section 5.1) is examined in greater depth, as it represents a balanced and realistic traffic condition where both vehicular and pedestrian flows are significant without reaching severe congestion. For the low- (Section 5.2) and high-demand (Section 5.3) scenarios, only global metrics — pedestrian and vehicle curves, average speeds, and phase distributions — are reported, as the behaviors and trends observed in the remaining metrics were consistent with those found in the mid-demand scenario. The halting metrics for both pedestrians and vehicles, as well as the speed metrics per intersection for the low- and high-demand scenarios, can be found in Appendix B. For these two demand levels, phase distributions were analyzed only for Strategy 1, while the corresponding results for Strategies 2 and 3 are also provided in Appendix B.

Table 5.1 summarizes where each performance metric is presented throughout the document, indicating the demand levels, control strategies, and corresponding sections or appendices in which the results can be found.

Table 5.1: Location of performance metrics across the document.

Metric	**Scenarios / Strategies Analyzed**	**Section / Appendix**
Pedestrian Curve	All	Section 5
Vehicle Curve	All	Section 5
Environment Average Speed	All	Section 5
Halting (Vehicles and Pedestrians)	Mid-demand Low-/High-demand	Section 5.1 Appendix B
Average Speed per Intersection	Mid-demand Low-/High-demand	Section 5.1 Appendix B
Average Phase Distribution	Mid-demand: all Strategies Low-/High-demand: Strategy 1 Low-/High-demand: Strategies 2–3	Section 5.1 Sections 5.2 / 5.3 Appendix B

5.1 Medium-Demand Scenario

5.1.1 Strategy 1

5.1.1.1 Pedestrian Curve

The pedestrian curve is a natural entry point, as it shows the temporal load of pedestrians present in the network. This curve provides context for interpreting halting: long or repeated pedestrian peaks tend to correlate with higher halting at specific nodes, while quick decay often implies fewer and shorter stops.

In Strategy 1, the three algorithms present a relatively similar behavior. QT-DQN shows a slight deviation caused by an isolated test run with poorer performance, which affected the average values. However, the median pedestrian count across all algorithms remains close to 220. DQN and MAPPO display almost identical curves, indicating that, in this scenario, both managed pedestrian flow with comparable efficiency.

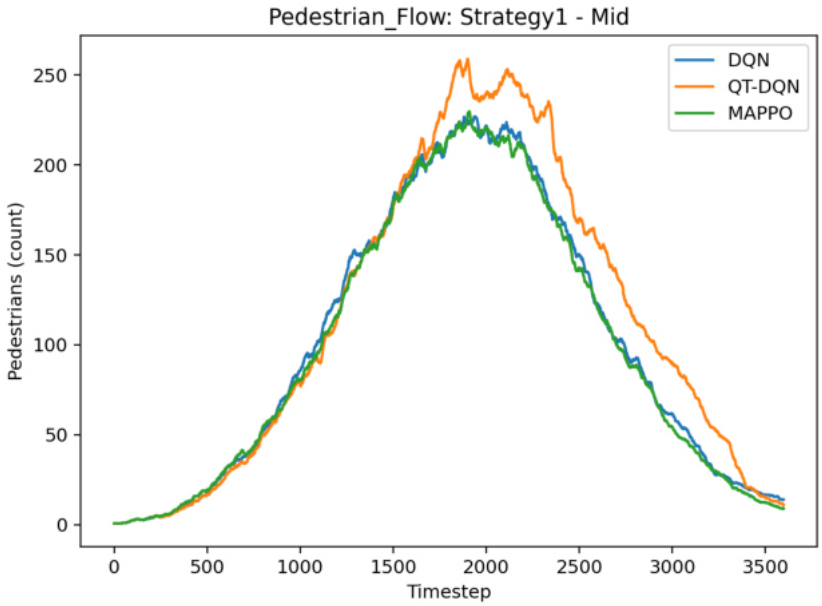


Figure 5.1: *Pedestrian Curve - Strategy 1, Mid scenario*

5.1.1.2 Vehicle Curve

The vehicle curve reveals how cars accumulate, how quickly they are cleared, and the total time needed to stabilize.

In this scenario, MAPPO reaches the peak earlier and starts decreasing promptly, while the DQN-based methods sustain a higher plateau before descending. Although DQN and QT-DQN are close to each other here, MAPPO remains more efficient at reducing the overall backlog and is the only algorithm capable of clearing all vehicles from the environment before the end of the period, demonstrating superior efficiency in maintaining smooth and continuous traffic flow.

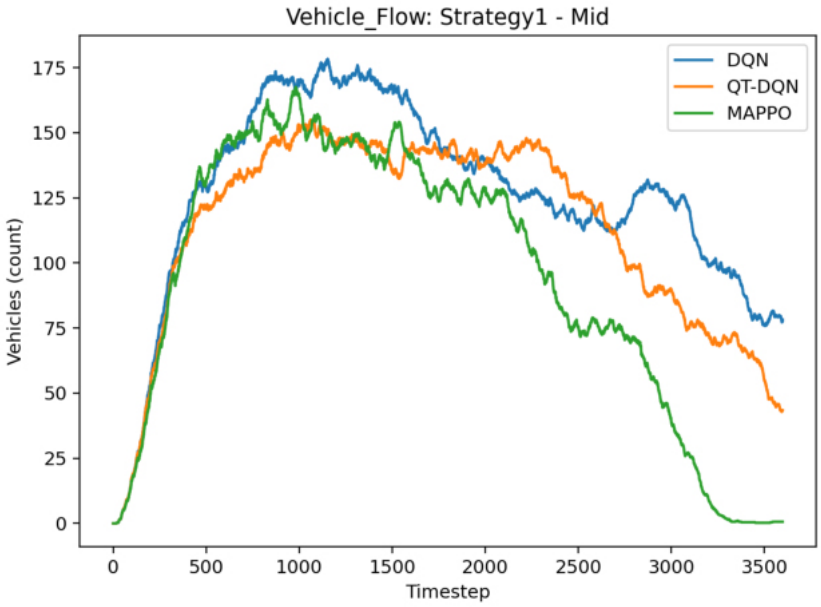


Figure 5.2: *Vehicle Curve - Strategy 1, Mid scenario*

5.1.1.3 Environment Average Speed

Environment average speed summarizes the overall mobility within the network. Consistent with the vehicle curve, MAPPO reaches higher speeds earlier. Also, MAPPO attains the maximum average speed before the end of the experiment, as all vehicles have already cleared the environment at that stage. This reflects faster stabilization and more efficient traffic evacuation. Both DQN variants maintain lower speeds and lag behind during the final third of the period, indicating slower recovery compared to MAPPO, although QT-DQN shows slightly better performance than standard DQN.

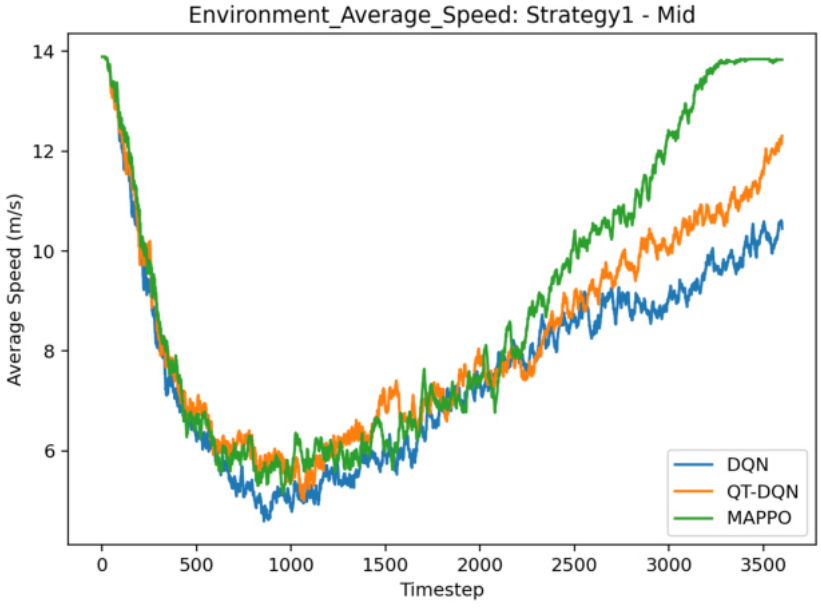
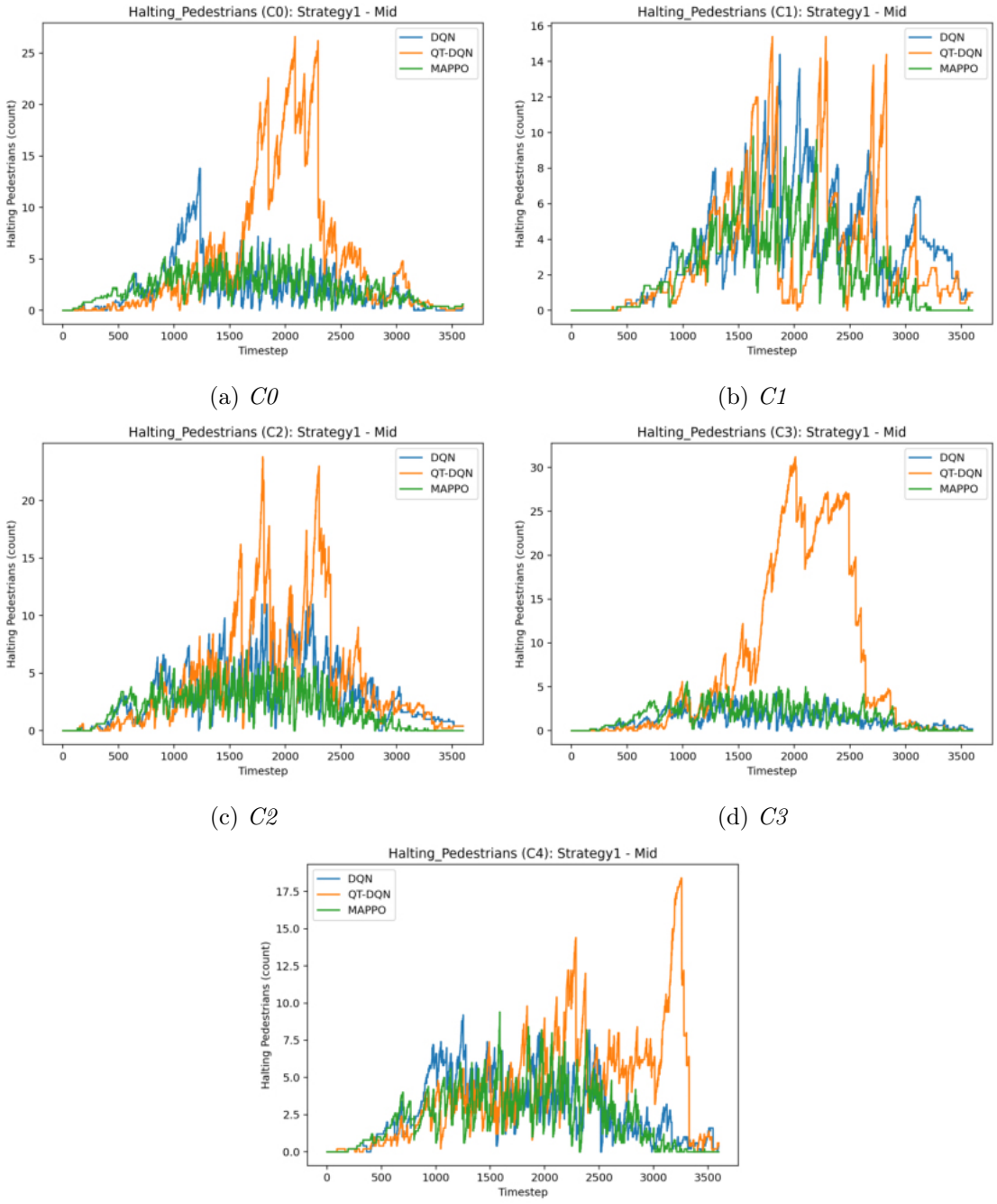


Figure 5.3: *Environment Average Speed - Strategy 1, Mid scenario*

5.1.1.4 Pedestrian Halting

Halting per intersection provides the local counterpart to the global flow view. MAPPO is the most stable, maintaining lower and more consistent pedestrian waiting times across intersections. It rarely exceeds seven pedestrians halted per intersection, which is a satisfactory outcome as it indicates effective control of pedestrian queues. The only exception is intersection C1, which is inherently more challenging as it concentrates both pedestrian and vehicle flows from all other intersections.



(a) *C0* (b) *C1*

(c) *C2* (d) *C3*

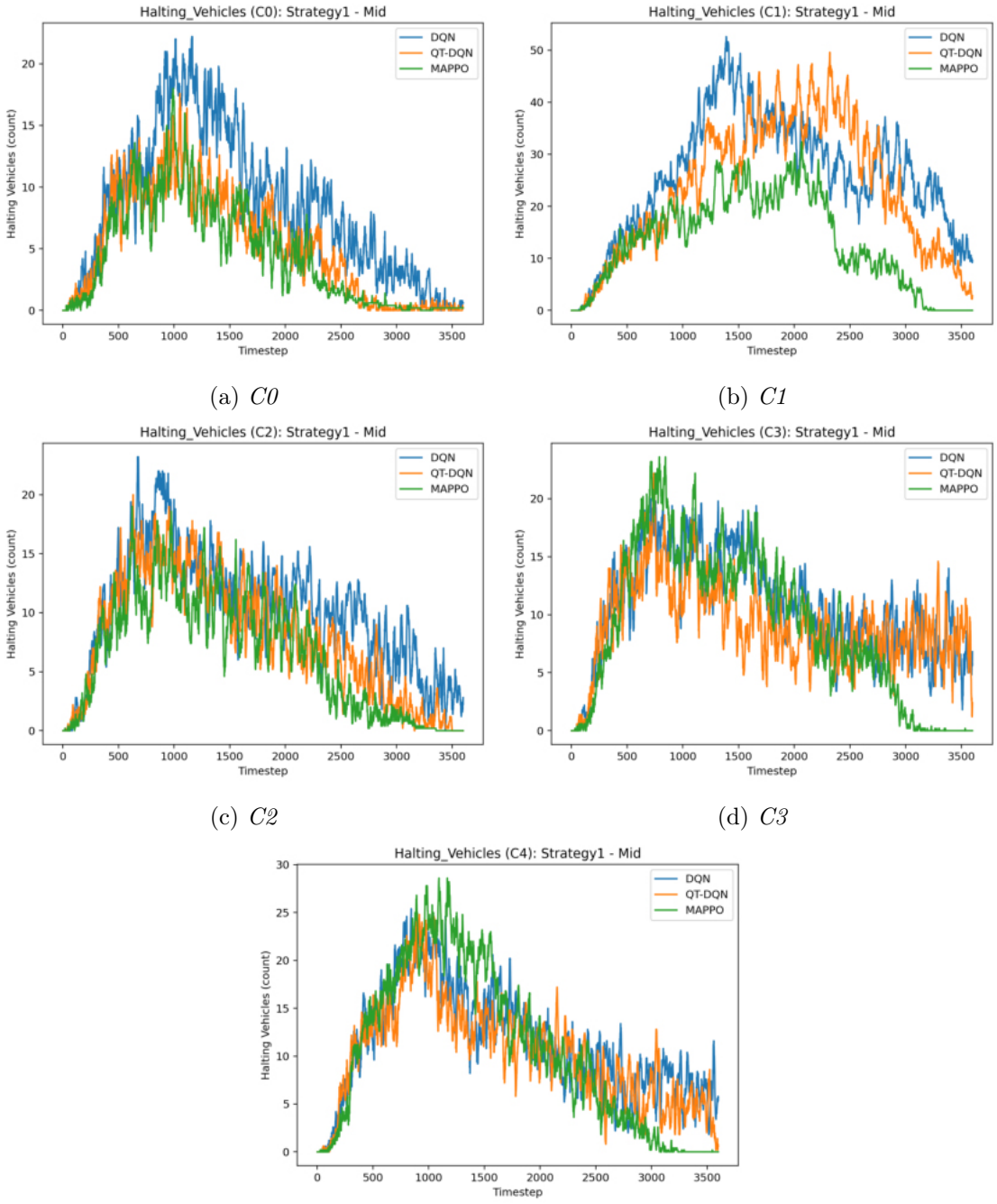
(e) *C4*

Figure 5.4: *Halting pedestrians per intersection - Strategy 1, Mid scenario*

DQN and QT-DQN, on the other hand, exhibit sporadic spikes (more frequent in QT-DQN), indicating occasional inconsistencies in pedestrian phase scheduling. This behavior in QT-DQN helps explain its poorer performance observed in Fig. 5.1, where pedestrian accumulation remains higher from the peak congestion period until the end of the period.

5.1.1.5 Vehicle Halting

The halting metric highlights how each algorithm manages vehicle coordination across each intersection.



(a) *C0*

(b) *C1*

(c) *C2*

(d) *C3*

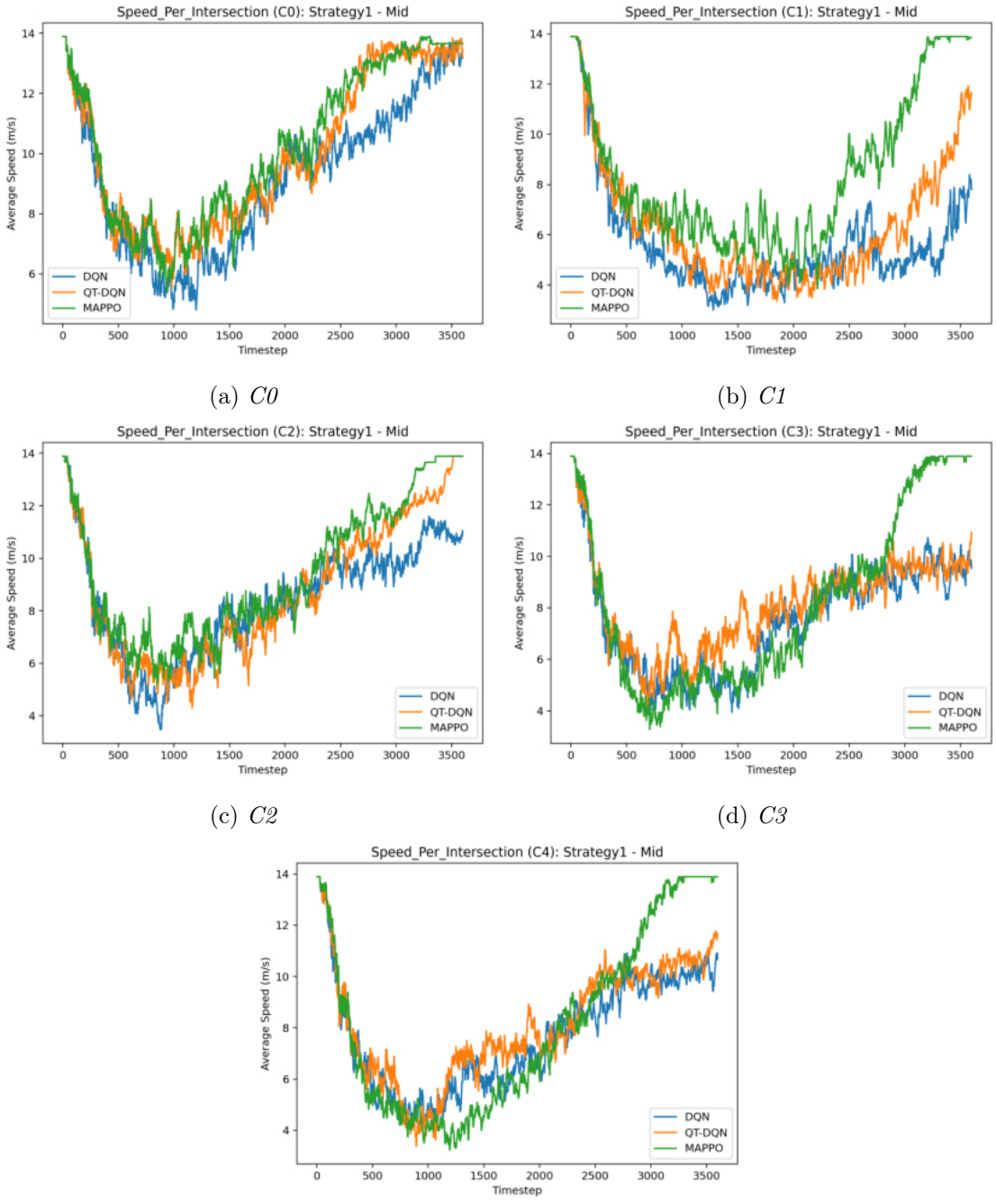
(e) *C4*

Figure 5.5: *Halting vehicles per intersection - Strategy 1, Mid scenario*

In Strategy 1, vehicle halting patterns are largely similar across algorithms during the first 2500 seconds, with no clear winner among the surrounding intersections (C0, C2, C3, C4). The initial phase corresponds to the accumulation period, where traffic density rises across the network and congestion is unavoidable regardless of control performance. In such conditions, the algorithms' capacity to prevent queuing is limited; instead, efficiency is measured by how quickly they can dissipate the resulting congestion once vehicle inflow stabilizes. From 2500 seconds onward, MAPPO clearly distinguishes itself, consistently maintaining lower halting levels across all intersections and clearing queues more rapidly than both DQN variants. This effect is particularly evident at the central intersection C1, the most influential node in the network, where MAPPO keeps halting counts steadily below those of DQN and QT-DQN, demonstrating superior coordination and recovery capability under balanced but high-interaction traffic conditions.

5.1.1.6 Average Speed per Intersection

The per-intersection average speed metric complements the halting analysis by reflecting the direct mobility outcome of vehicle coordination. As expected, intersections with higher halting counts correspond to lower average speeds. Following the trends observed previously, MAPPO consistently maintains superior performance, sustaining higher speeds across all intersections, particularly at the central node C1. This pattern mirrors its lower halting levels and faster queue dissipation observed after 2500 seconds.



(a) *C0*

(b) *C1*

(c) *C2*

(d) *C3*

(e) *C4*

Figure 5.6: *Average speed per intersection - Strategy 1, Mid scenario*

As shown in Fig. 5.3, MAPPO is also the only algorithm that reaches near-maximum average speeds toward the end of the period, which occurs once the environment is almost entirely cleared of vehicles, as evidenced in Fig. 5.5.

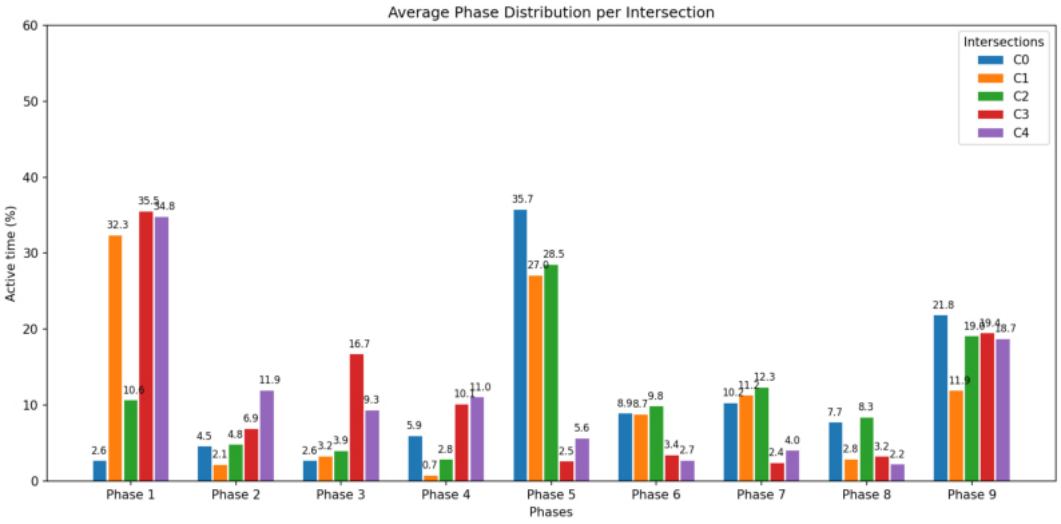
Between the DQN-based methods, differences are less pronounced; however, QT-DQN tends to achieve slightly higher speeds, especially at intersections C1, C0, and C2, suggesting marginally better coordination in these areas.

5.1.1.7 Average Phase Distribution

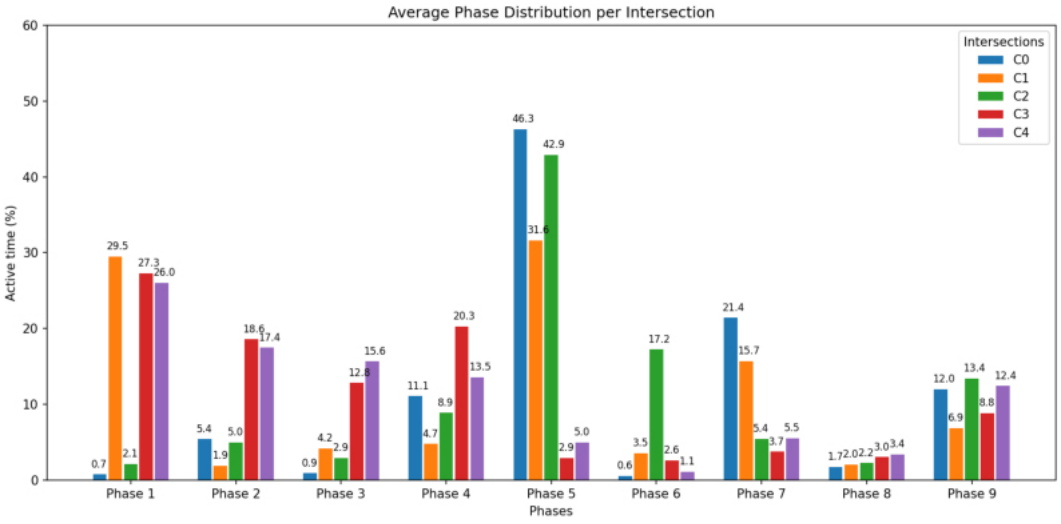
Phase distributions (see Section 3.2.1) provide insight into how each algorithm allocates signal time across movements and intersections. It is important to recall that in the DQN-based methods all intersections share the same network weights and receive no information about which intersection they control, whereas in MAPPO a shared network is used, but each agent receives a one-hot encoded (OHE) identifier, enabling it to adapt its policy to the specific intersection context. This distinction allows MAPPO agents to specialize their behavior according to local traffic conditions.

As a result, the DQN algorithms exhibit greater variability and less consistent phase selection patterns, while MAPPO demonstrates more stable and specific choices. Notably, MAPPO rarely activates the left-turn phases (4 and 8), instead focusing on phases that maximize vehicle clearance from the environment. At peripheral intersections, this specialization becomes evident: C0 predominantly selects phases 5 and 7 (W–E movements), C2 favors phases 5 and 6 (W–E and westbound), C3 prioritizes phases 1 and 3 (N–S and southbound), and C4 emphasizes phases 1 and 2 (N–S and northbound). This behavior is particularly interesting, as such targeted phase selection had not been as clearly observed in the DQN-based approaches.

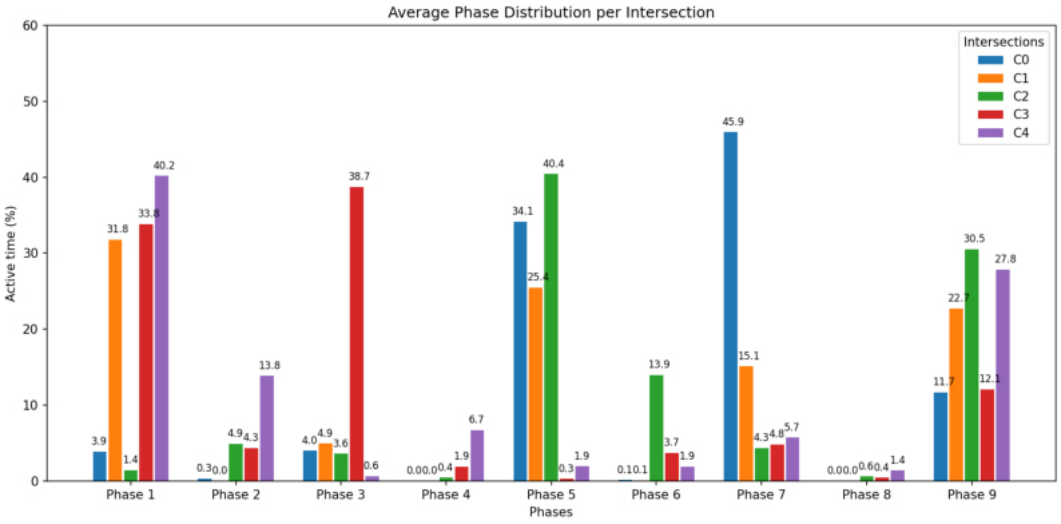
Additionally, QT-DQN appears to underutilize phase 9 compared to the other algorithms, which may help explain its poorer pedestrian coordination performance observed in Fig. 5.4. Overall, these results highlight the advantage of MAPPO's intersection-aware policy representation, enabling more adaptive and context-specific phase control under Strategy 1.



(a) *DQN*



(b) *QT-DQN*



(c) *MAPPO*

Figure 5.7: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 1, Mid scenario).*

5.1.2 Strategy 2

5.1.2.1 Pedestrian Curve

With Strategy 2's more homogeneous flows along the radial axis, just like in Strategy 1, DQN and MAPPO exhibit similar performance, maintaining relatively stable pedestrian flows throughout time. However, QT-DQN once again shows poor behavior, with noticeably higher pedestrian accumulation. This suggests that the algorithm occasionally fails to account for pedestrian phases, leading to longer waiting times.

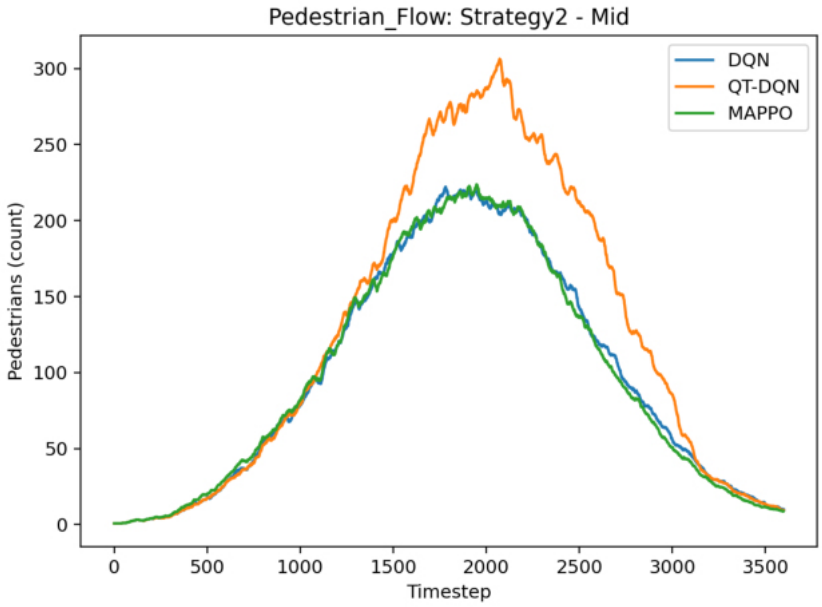


Figure 5.8: *Pedestrian Curve - Strategy 2, Mid scenario*

5.1.2.2 Vehicle Curve

Homogeneity favors all methods, narrowing performance gaps relative to Strategy 1 in the final third. As observed previously in Fig. 5.2, MAPPO still declines faster from peak accumulation and clears vehicles earlier. DQN and QT-DQN closely track each other, although QT-DQN shows a slight advantage in vehicle clearance over standard DQN.

Unlike in strategy 1, the total number of vehicles never reaches zero by the end of the simulation run, since most vehicles are generated along the radial corridor (at intersections C3 and C4) and fewer lanes are available for insertion. As a result, the simulator requires more time to inject all scheduled vehicles, and the process is not completed before the period ends. Consequently, even when the network is nearly empty, a small number of vehicles remain present in the environment.

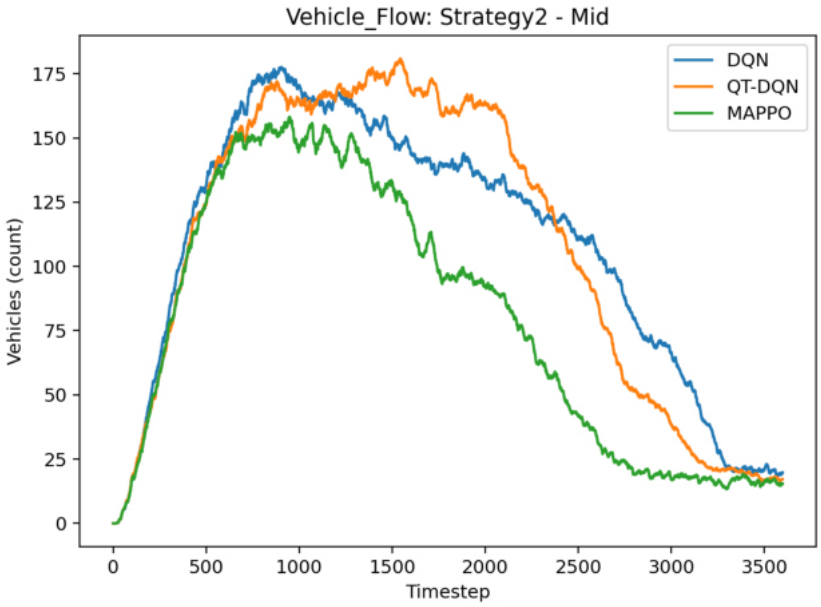


Figure 5.9: *Vehicle Curve - Strategy 2, Mid scenario*

5.1.2.3 Environment Average Speed

Average speed confirms the previous observations, showing a behavior similar to that seen in Fig. 5.3. In this case, MAPPO separates from the DQN-based methods earlier and consistently maintains the highest speeds until the end of the window. Both DQN and QT-DQN benefit from the homogeneity of the strategy, reaching maximum speeds in the final stages and displaying overall more satisfactory performance compared to Strategy 1. However, the maximum speeds never fully stabilize due to the continuous injection of vehicles throughout the simulation.

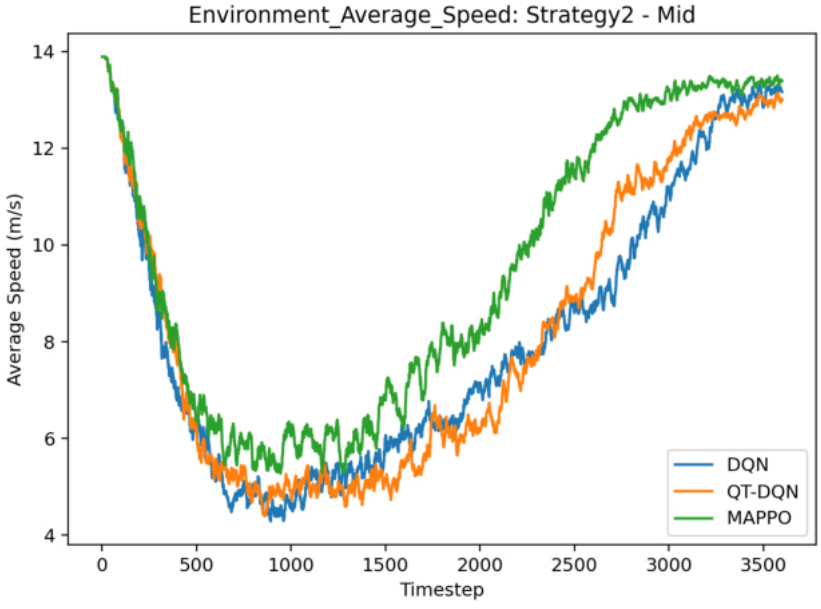
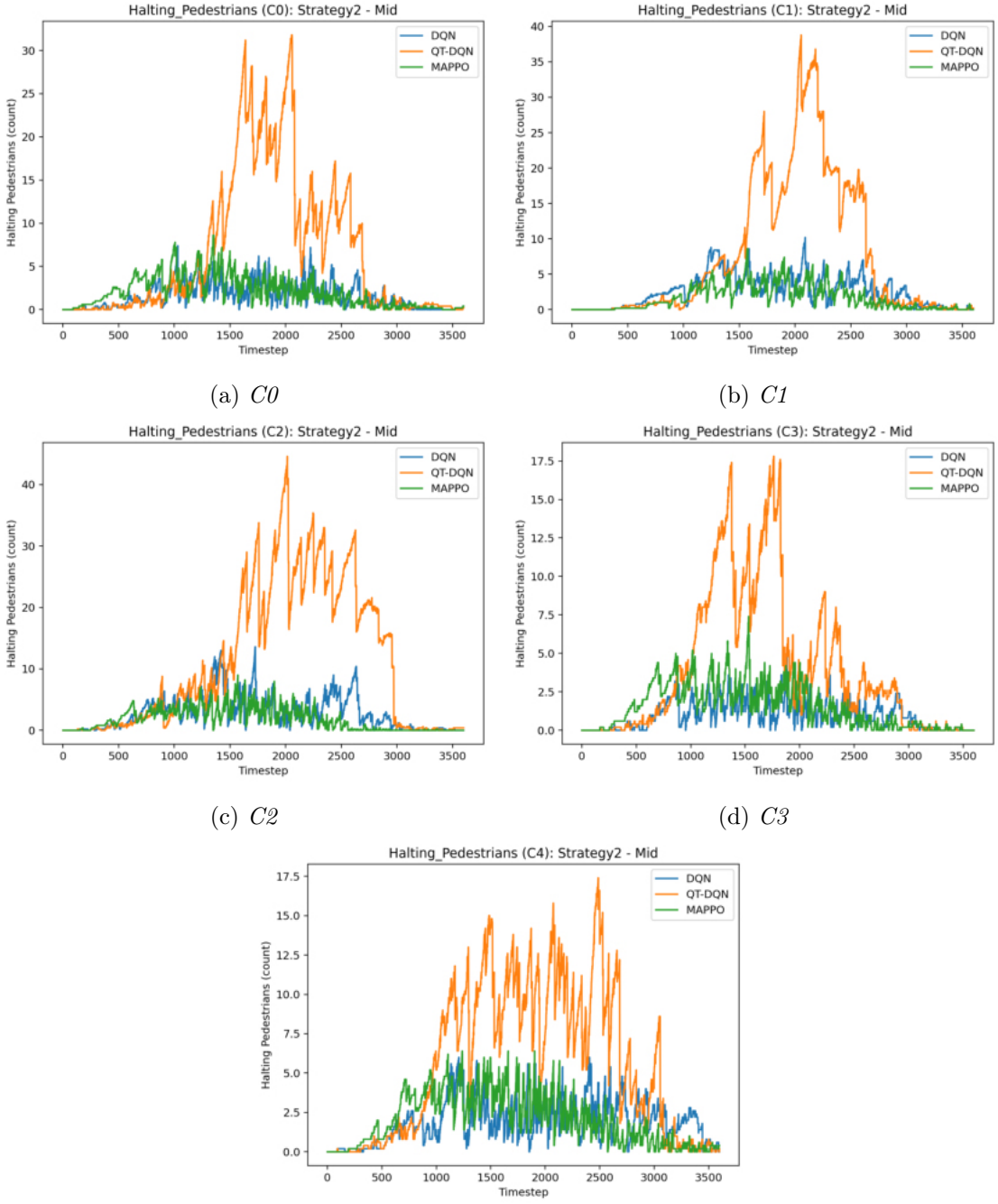


Figure 5.10: *Environment Average Speed - Strategy 2, Mid scenario*

5.1.2.4 Pedestrian Halting

It is worth noting that the traffic strategies do not directly influence pedestrian flow, as vehicle-generation patterns only affect pedestrian phases indirectly through signal scheduling. In principle, the higher vehicle density at intersections C3 and C4 could have slightly disadvantaged pedestrian movement by reducing green-time availability, but this effect was not observed in the results.



(a) *C0*

(b) *C1*

(c) *C2*

(d) *C3*

(e) *C4*

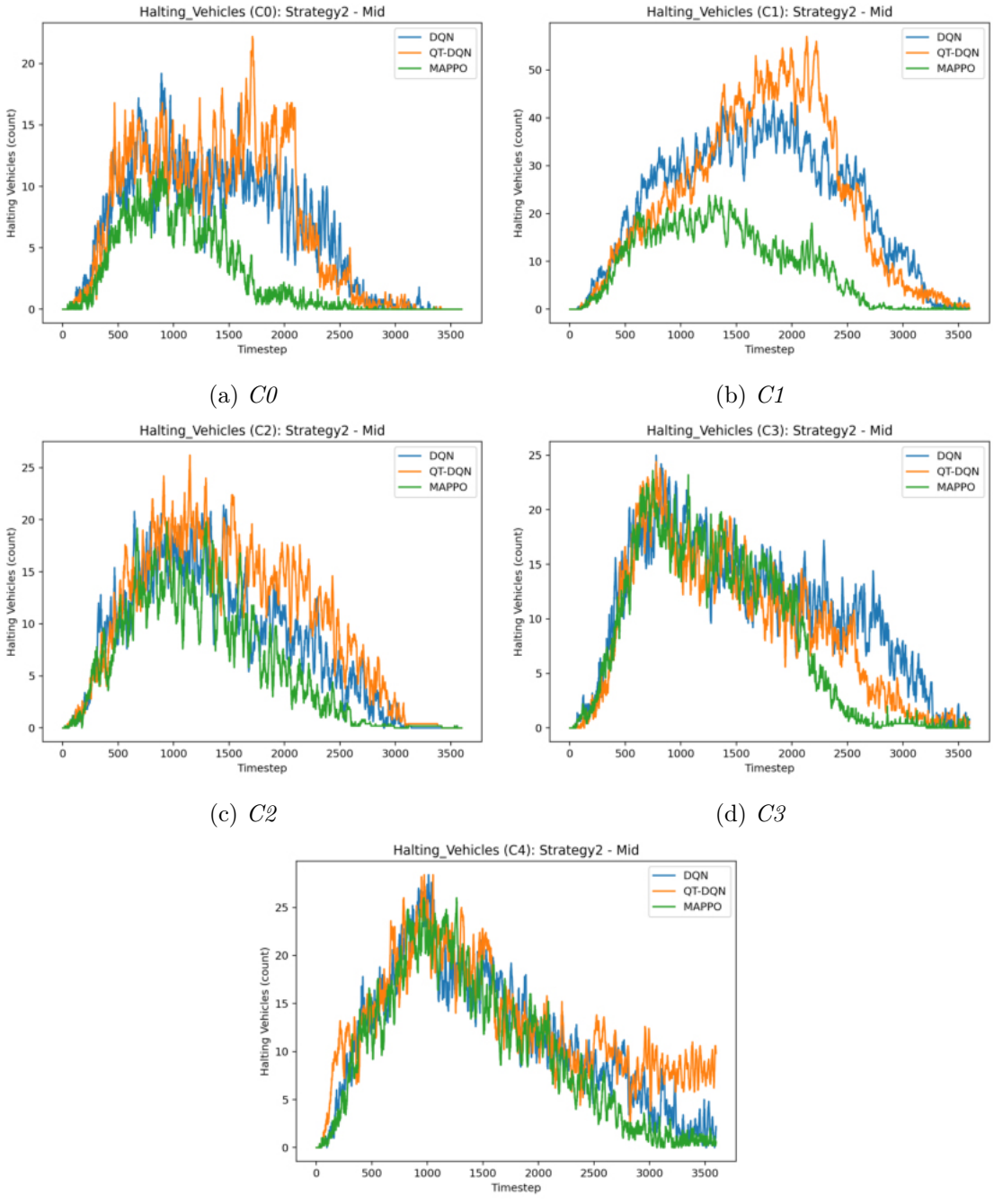
Figure 5.11: *Halting pedestrians per intersection - Strategy 2, Mid scenario*

With fewer conflicting maneuvers, pedestrian halting remains low and stable across all

intersections in both DQN and MAPPO, achieving satisfactory performance and maintaining similarly low pedestrian waiting times even at the central intersection C1. QT-DQN, however, again shows weaker behavior, occasionally disregarding pedestrian phases and leading to higher halting levels. This behavior in QT-DQN may suggest a potential limitation: by incorporating the influence of all neighbors with the same weighting factor β, the algorithm ignores that in homogeneous traffic conditions, certain intersections can exert more influence than others depending on their specific flow patterns.

5.1.2.5 Vehicle Halting

Strategy 2 eases overall coordination, and MAPPO capitalizes on it, minimizing stops consistently across all intersections. DQN and QT-DQN converge closely like in Strategy 1, yet still exhibit localized peaks, particularly at intersections C0 and C1. As expected, both DQN-based methods also experience greater difficulty at intersections C3 and C4, where vehicle inflow is higher along the vertical axis, leading to increased halting levels. In contrast, MAPPO maintains uniform performance throughout the network, effectively handling the heavier radial demand.



(a) *C0*

(b) *C1*

(c) *C2*

(d) *C3*

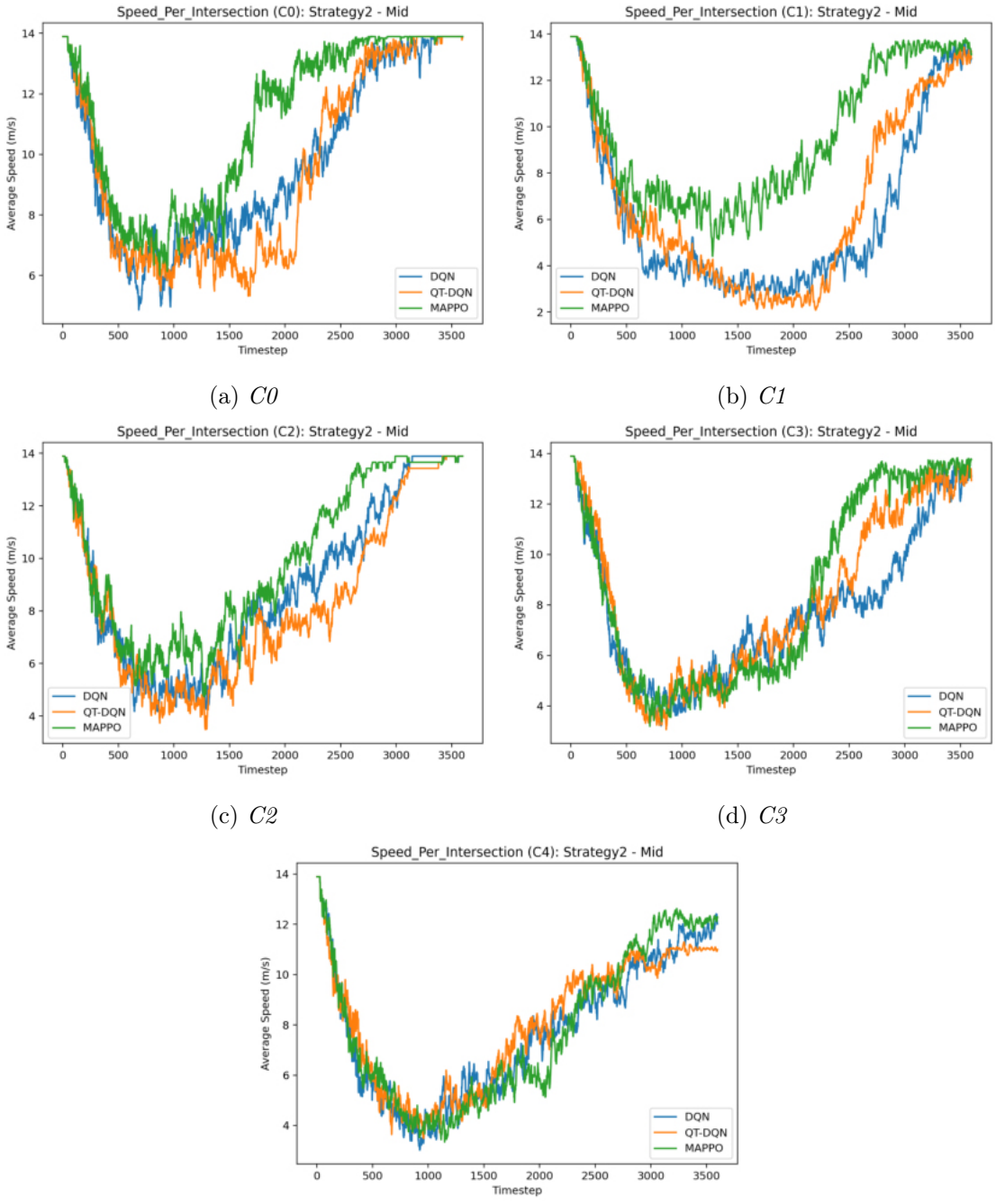
(e) *C4*

Figure 5.12: *Halting vehicles per intersection - Strategy 2, Mid scenario*

The results at intersection C1 are particularly noteworthy, where MAPPO achieves excellent stability and minimal halting, reinforcing its superior coordination capability.

5.1.2.6 Average Speed per Intersection

Intersection speeds confirm the same narrative. MAPPO sustains higher averages, notably at C1, while DQN and QT-DQN remain close but consistently lower. All methods reach their maximum average speeds before the end of the window, except at intersection C4, where vehicles keep being injected along the radial corridor — predominantly moving northbound — preventing full recovery to free-flow conditions.



(a) *C0*

(b) *C1*

(c) *C2*

(d) *C3*

(e) *C4*

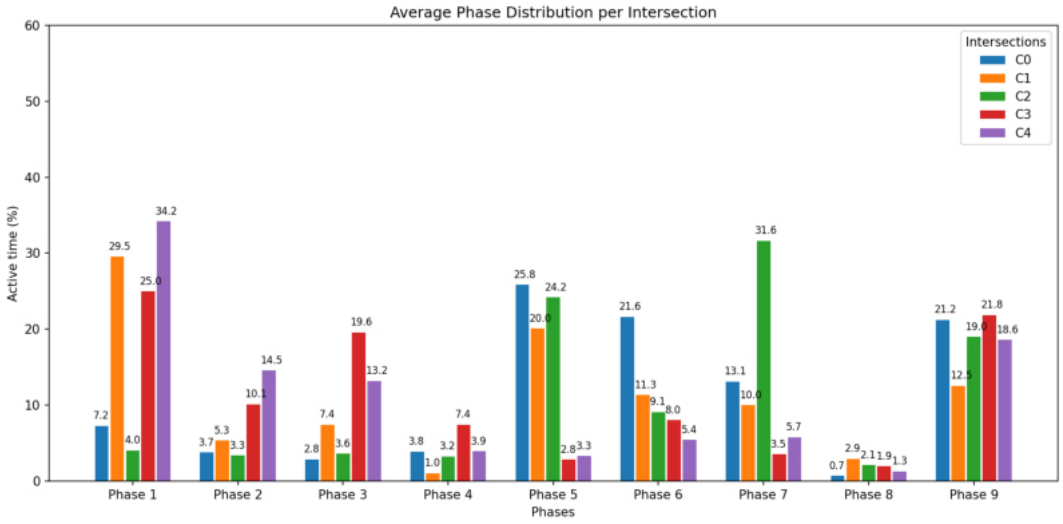
Figure 5.13: *Average speed per intersection - Strategy 2, Mid scenario*

5.1.2.7 Average Phase Distribution

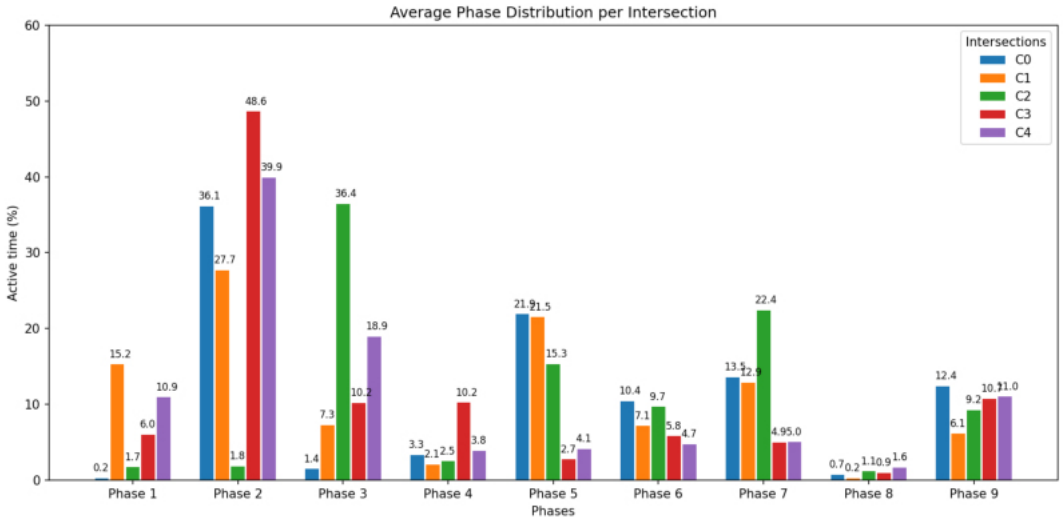
Phase usage becomes more structured under Strategy 2. MAPPO's allocation remains highly demand-aware, prioritizing radial flows when necessary without neglecting cross movements or pedestrians. DQN and QT-DQN display less balanced behavior compared to Strategy 1, and all methods continue to use left-turn phases 4 and 8 very rarely.

QT-DQN, in particular, is less fair to pedestrians, as its phase scheduling is inconsistent and occasionally fails to allocate pedestrian phases properly, which aligns with its higher halting levels observed earlier.

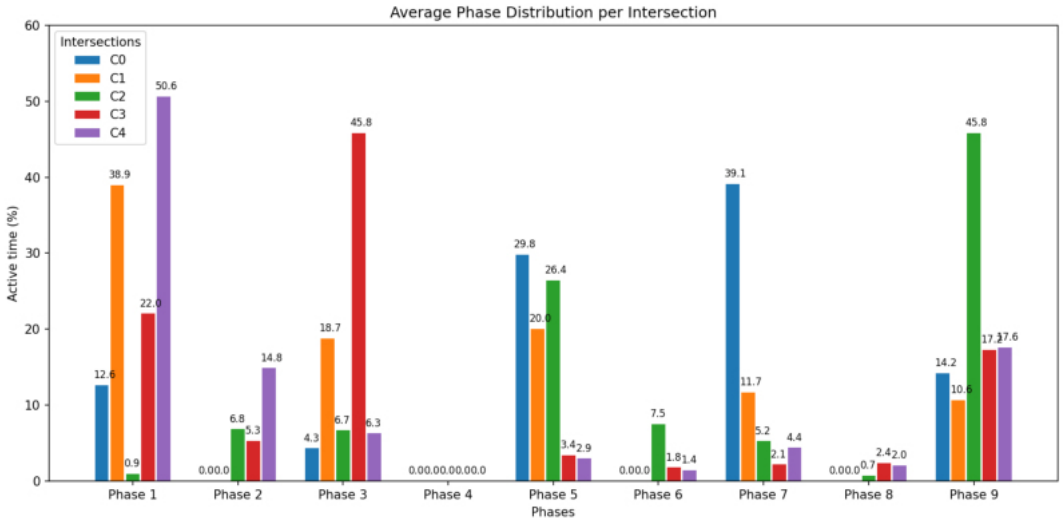
MAPPO, on the other hand, exhibits once again a clear and purposeful phase pattern across intersections. At C0, phases 5 and 7 (west–east movements) are preferred, while C2 favors phase 5; unlike in Strategy 1, it uses phase 6 less frequently and allocates more time than expected to phase 9, although this did not compromise overall performance. The central intersection C1 concentrates on the dominant arterial flows, selecting mainly phases 1 and 5, along with phase 3 to support northbound clearance. At C3, the policy remains strongly centered on phases 1 and 3, as expected, whereas C4 effectively resolves its previous coordination issues by emphasizing phase 1. Once again, phases 4 and 8 see minimal activation, along with phase 6, which could have been expected to appear more often at C2.



(a) *DQN*



(b) *QT-DQN*



(c) *MAPPO*

Figure 5.14: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 2, Mid scenario).*

5.1.3 Strategy 3

5.1.3.1 Pedestrian Curve

Strategy 3 exhibits similar pedestrian loads across algorithms, with small deviations stemming from indirect interference of vehicle phases. As before, this curve is mainly a context setter for halting: longer or recurrent peaks generally forecast more pedestrian waiting at specific intersections.

Once again, DQN and MAPPO display very similar behavior, maintaining stable pedestrian counts throughout time, while QT-DQN continues to show irregularities associated with weaker pedestrian-phase management. Such discrepancies are expected to be more evident under the mid-demand scenario, where the balance between vehicle and pedestrian flows makes coordination errors more visible. In contrast, under low demand these issues tend to be attenuated due to lighter traffic, and under high demand they become less distinguishable, as saturation effects dominate and all methods inevitably accumulate more elements within the intersections.

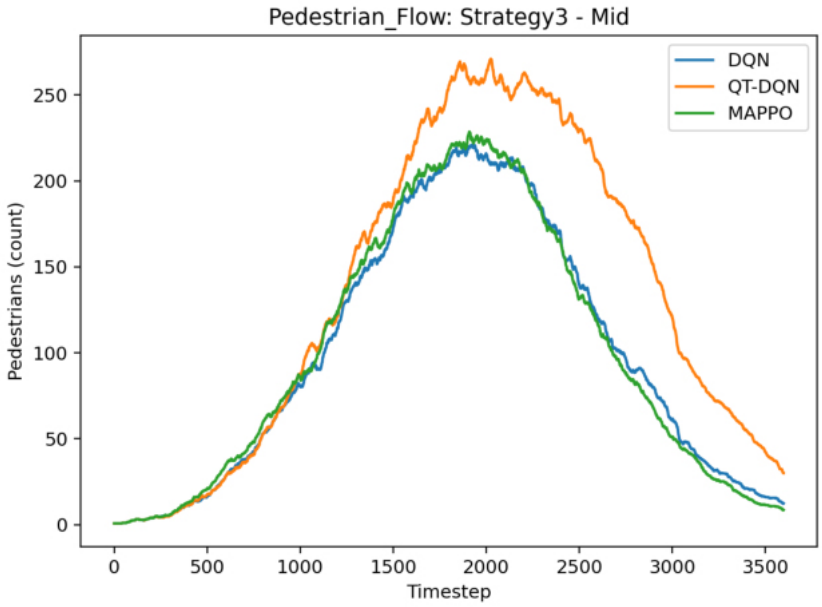


Figure 5.15: *Pedestrian curve - Strategy 3, Mid scenario*

5.1.3.2 Vehicle Curve

As in Strategy 2, the more structured flows benefit all methods. MAPPO retains the lead: although it may appear slower during the first half, when sequencing phases to balance queues, it compensates later, clearing vehicles much sooner and reducing overall congestion. DQN and QT-DQN follow similar trajectories, with QT-DQN initially performing slightly better and never exceeding 150 vehicles in the network. However, toward the end of the window, QT-DQN falls behind, proving less effective at clearing the remaining traffic, while standard DQN achieves a slightly better final clearance. Overall, MAPPO maintains a clear advantage in both recovery speed and completeness of vehicle evacuation.

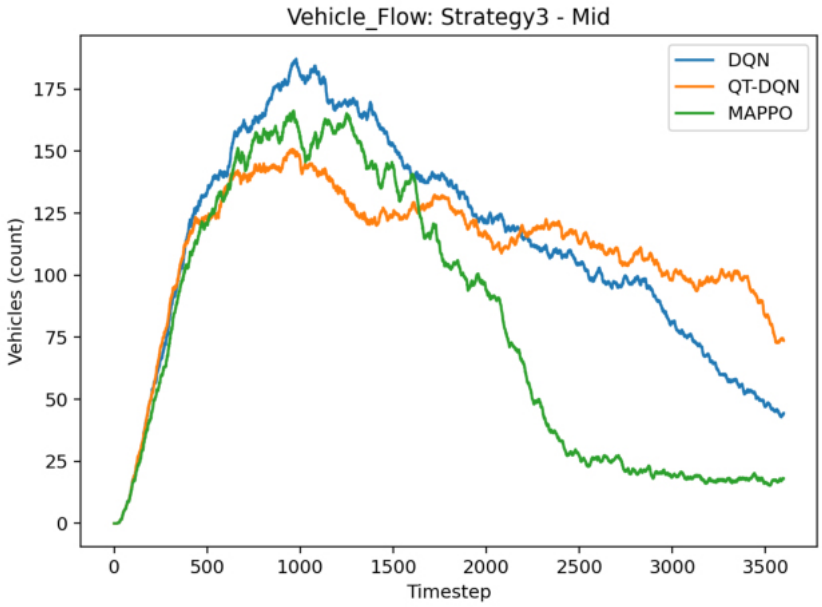


Figure 5.16: *Vehicle Curve - Strategy 3, Mid scenario*

5.1.3.3 Environment Average Speed

Global speed matches the vehicle-curve reading: MAPPO achieves higher late-episode speeds, signaling earlier clearance of remaining queues. DQN/QT-DQN remain close for much of the simulation period, but the closing gap reappears when the network must finish emptying.

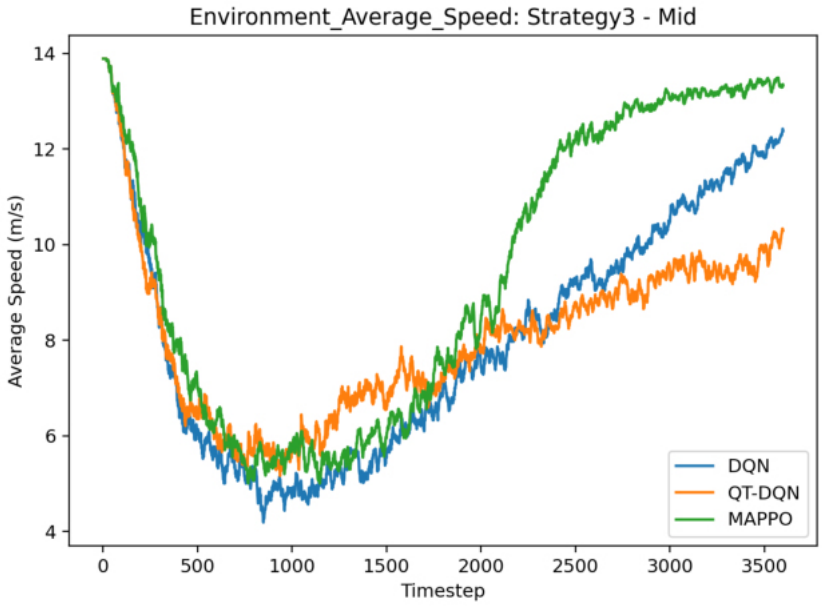
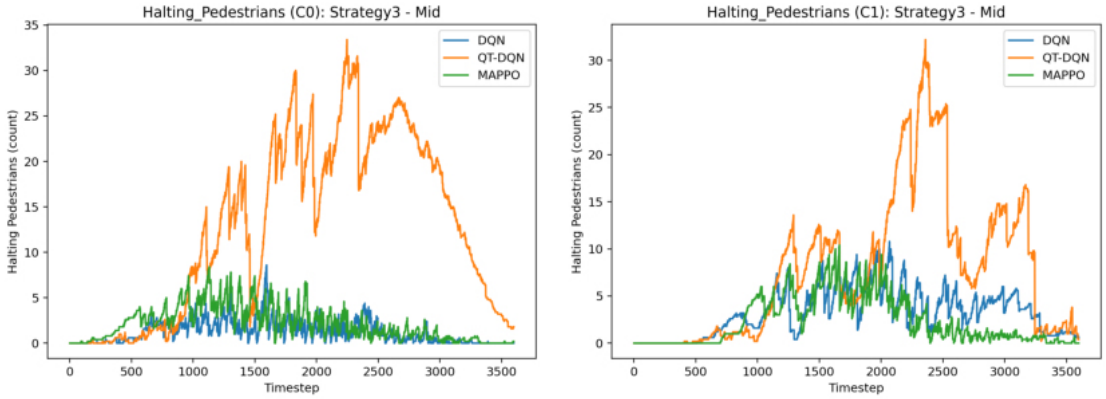


Figure 5.17: *Environment Average Speed - Strategy 3, Mid scenario*

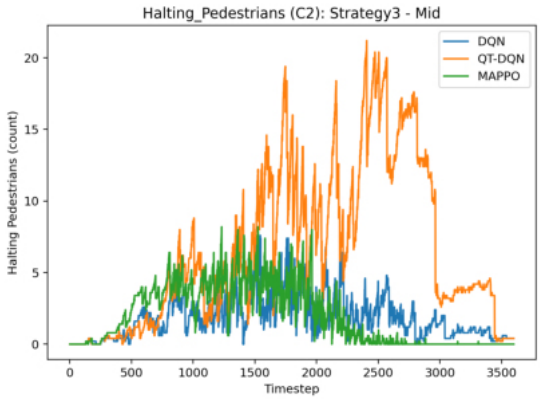
5.1.3.4 Pedestrian Halting

Results under Strategy 3 closely mirror those observed in Strategy 2. Both DQN and MAPPO maintain low and stable pedestrian halting levels across all intersections, demonstrating consistent coordination between vehicle and pedestrian phases. QT-DQN, however, once again exhibits strong oscillations and accumulates a significantly higher number of pedestrians at every intersection, confirming its recurrent difficulty in managing pedestrian phases effectively.

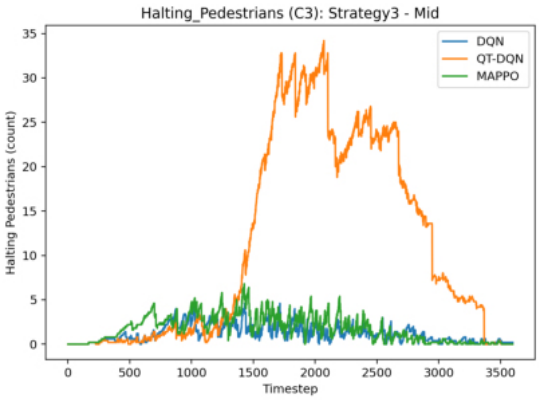


(a) *C0*

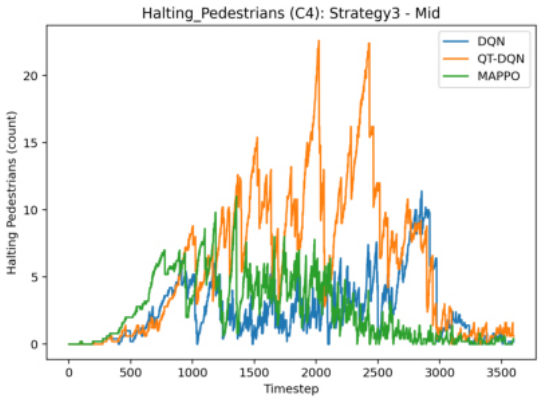
(b) *C1*



(c) *C2*



(d) *C3*

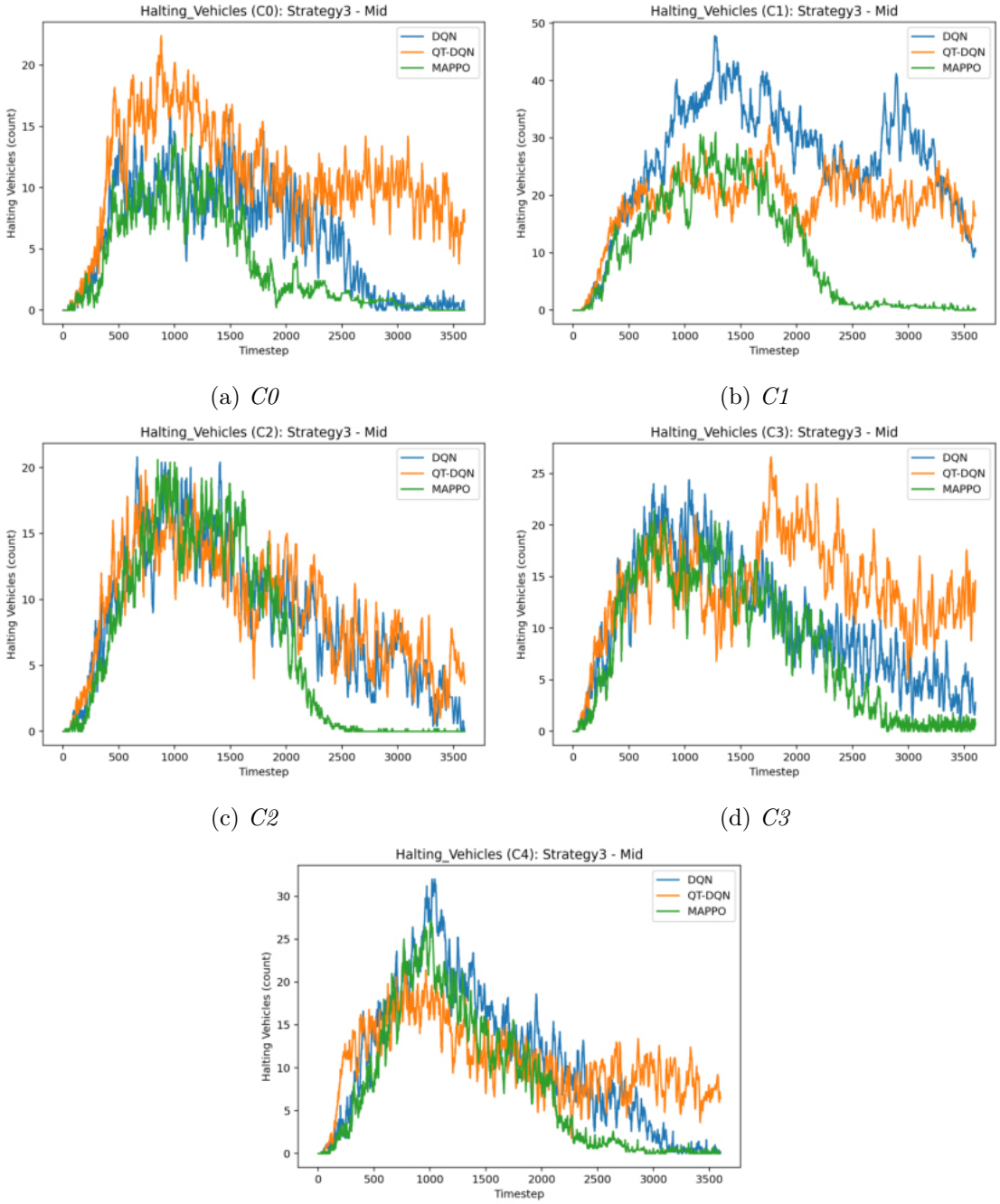


(e) *C4*

Figure 5.18: *Halting pedestrians per intersection - Strategy 3, Mid scenario*

5.1.3.5 Vehicle Halting

MAPPO once again delivers the best overall performance, maintaining the lowest halting levels across all intersections. Its advantage is particularly evident at the central intersection C1 and at C3, where most vehicles enter the network. Remarkably, MAPPO is the only algorithm capable of reducing vehicle halting to zero at all intersections by the end of the episode.



(a) *C0*

(b) *C1*

(c) *C2*

(d) *C3*

(e) *C4*

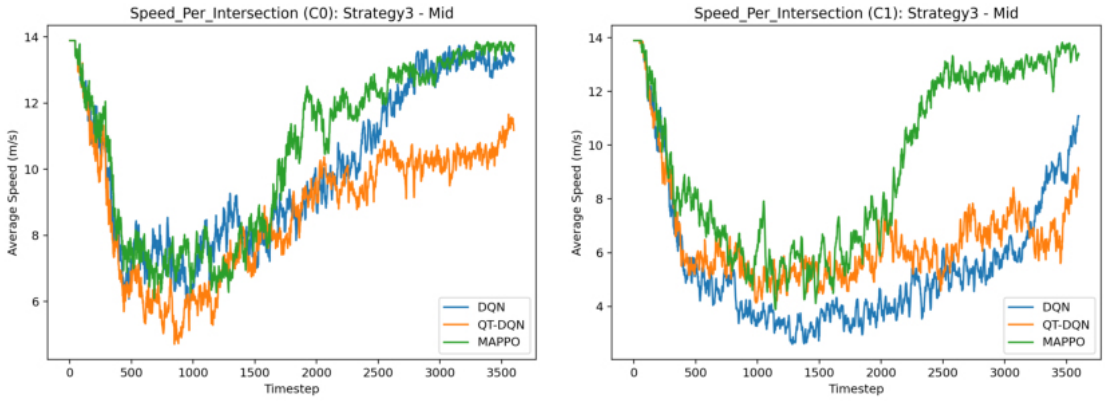
Figure 5.19: *Halting vehicles per intersection - Strategy 3, Mid scenario*

An interesting observation emerges from these results. Although Strategies 2 and 3 share

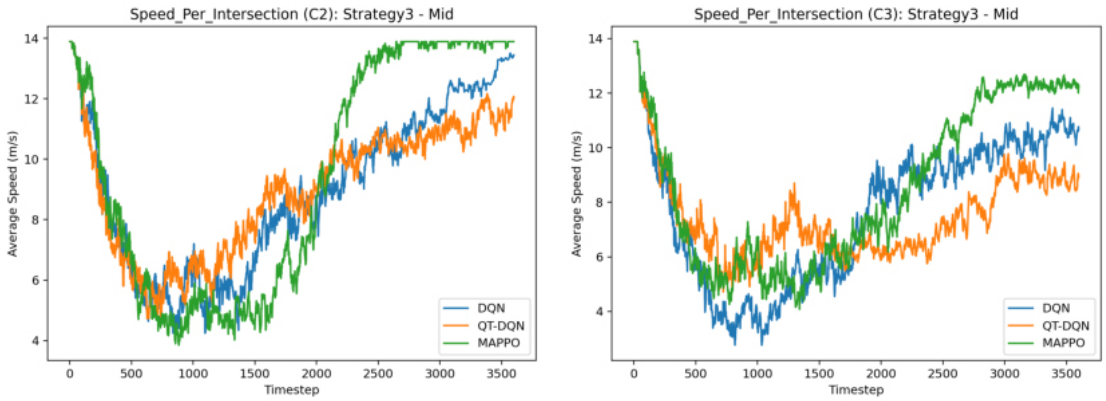
nearly identical configurations — differing only in the direction of the main radial flow — the relative behavior of the DQN-based methods changes substantially between them. While QT-DQN appeared slightly superior to DQN in Strategy 2, it performs noticeably worse in Strategy 3, exhibiting higher variability and less reliable congestion management. This reinforces the inherent instability and unpredictability of Deep Q-Learning methods when applied to complex multi-agent traffic environments: performance cannot be assumed consistent across scenarios or trainings. In contrast, MAPPO demonstrates stable and robust behavior in all intersections and strategies, confirming its superior adaptability and coordination capability.

5.1.3.6 Average Speed per Intersection

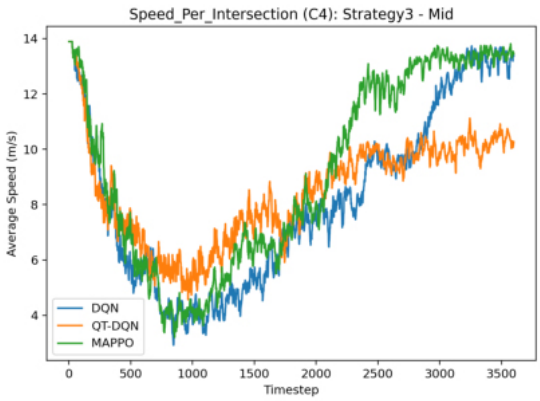
Results follow the same pattern observed in the vehicle halting analysis. MAPPO sustains the highest average speeds across all intersections, reflecting its faster and more complete clearance of traffic. DQN and QT-DQN show comparable but consistently lower values, in line with their higher residual halting levels. At intersection C3, the maximum average speed is never reached, as the same limitation identified in Strategy 2 persists: continuous vehicle injection along the radial corridor prevents the network from fully stabilizing before the end of the period.



(a) *C0* (b) *C1*



(c) *C2* (d) *C3*



(e) *C4*

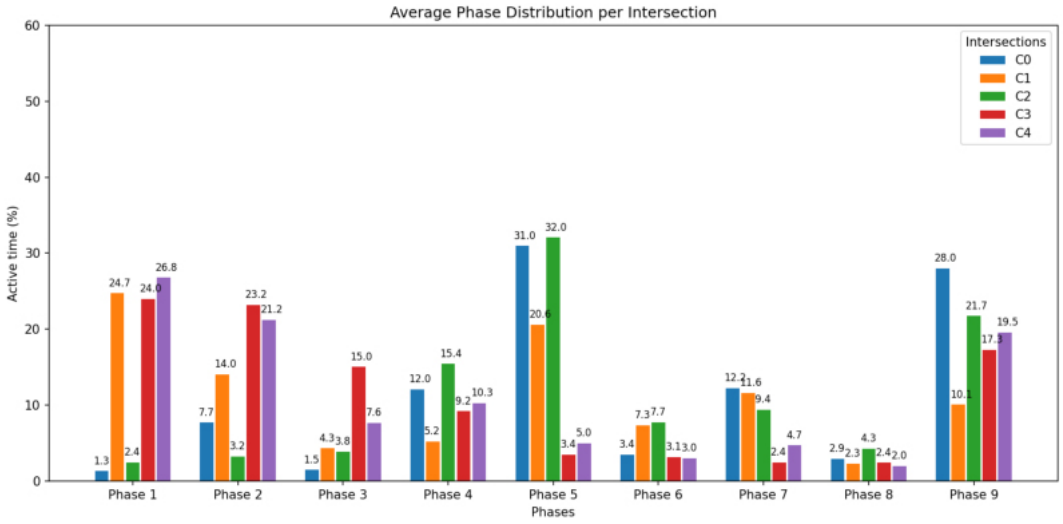
Figure 5.20: *Average speed per intersection - Strategy 3, Mid scenario*

5.1.3.7 Average Phase Distribution

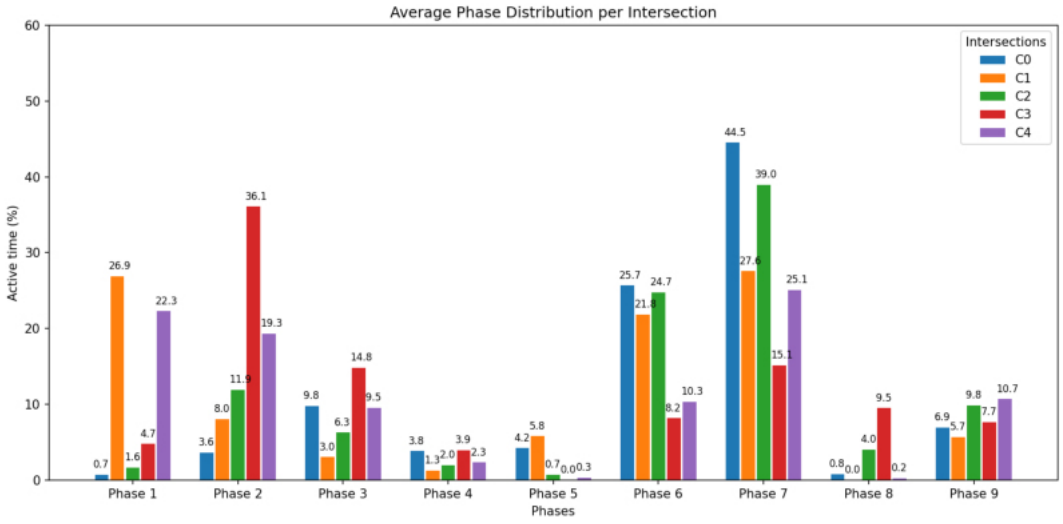
The phase usage patterns under Strategy 3 reveal distinct behavioral profiles among the algorithms. DQN presents a very balanced and varied distribution, with all phases being used across all intersections. This uniformity, however, highlights its limited ability to adapt to the more homogeneous flow structure of Strategy 3, where intersections C3 and C4 predominantly handle southbound traffic, in contrast to C0 and C2. Nevertheless, DQN shows an effective utilization of phase 9, ensuring satisfactory pedestrian service.

QT-DQN behaves more selectively but again underestimates the importance of phase 9, which negatively impacts pedestrian coordination, consistent with the higher halting levels observed earlier.

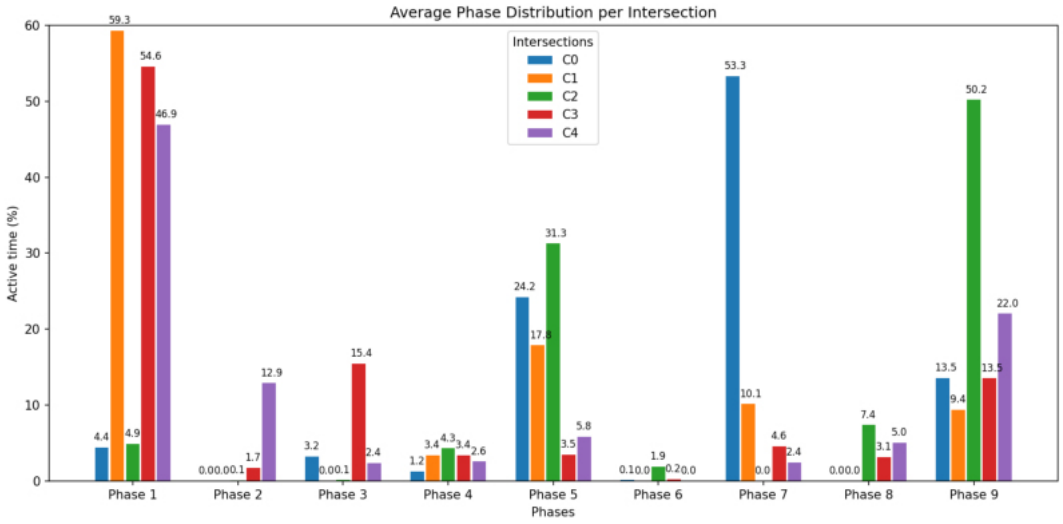
MAPPO, in contrast, exhibits much more decisive and context-aware phase selection. At C0, it primarily uses phase 7 (east-to-all), effectively managing the high inflow of vehicles approaching from the east regardless of their destination. Intersection C1 once again relies mostly on the dominant arterial phases 1 and 5; interestingly, phase 2 (north-to-all) is barely used, but this is compensated by the frequent activation of phase 1, which serves a similar purpose. At C2, MAPPO prioritizes phases 5 and 8 to handle radial flows, complemented by phases 1 and 4 for vehicles arriving from the opposite axis. Intersection C3 focuses on phases 1 and 3 to process both the heavy southbound inflow and vehicles received from C1, while C4 predominantly activates phases 1 and 2, as expected. Overall, MAPPO's choices are highly consistent with traffic demand and reflect an efficient and localized adaptation strategy.



(a) *DQN*



(b) *QT-DQN*



(c) *MAPPO*

Figure 5.21: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 3, Mid scenario).*

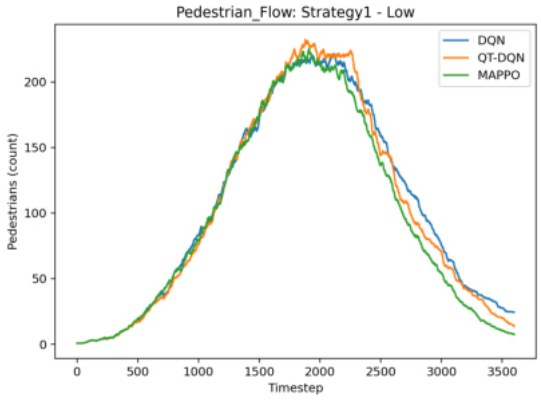
5.2 Low-Demand Scenario

5.2.1 Strategy 1

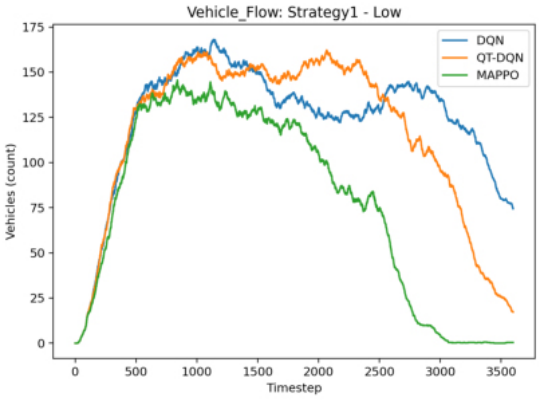
5.2.1.1 Pedestrian Curve, Vehicle Curve and Environment Average Speed

Under low-demand conditions, all algorithms are expected to maintain the network close to free-flow operation.

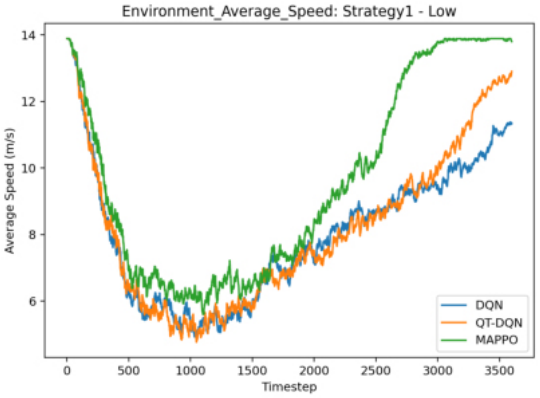
The pedestrian curves are nearly identical across the three methods, converging to a total of around 220 pedestrians, which indicates that the lighter traffic load allows all controllers to serve pedestrians efficiently. In the vehicle and environment speed curves, differences become slightly more noticeable. QT-DQN performs somewhat better than standard DQN, maintaining lower vehicle accumulation and higher average speeds in the final stage. However, MAPPO remains the most effective, being the only algorithm capable of fully clearing the network by the end of the simulation, even under low demand.



(a) *Pedestrian Curve*



(b) *Vehicle Curve*



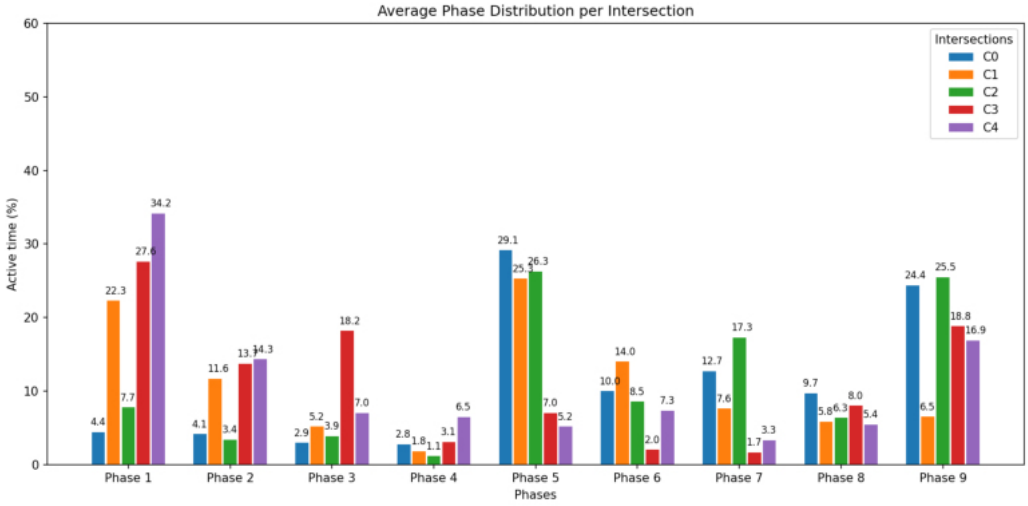
(c) *Environment Average Speed*

Figure 5.22: *Global metrics (Strategy 1, Low scenario).*

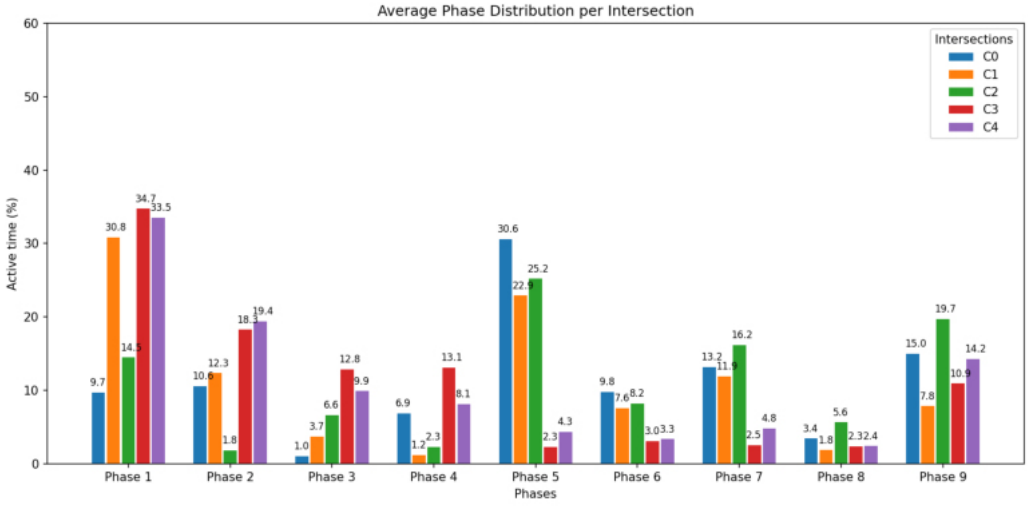
5.2.1.2 Average Phase Distribution

Phase distributions under low demand are very similar to those observed in Fig. 5.7, following the same expected patterns described previously. The main difference relative to the mid-demand case is the higher allocation to phase 9 across all algorithms, which results

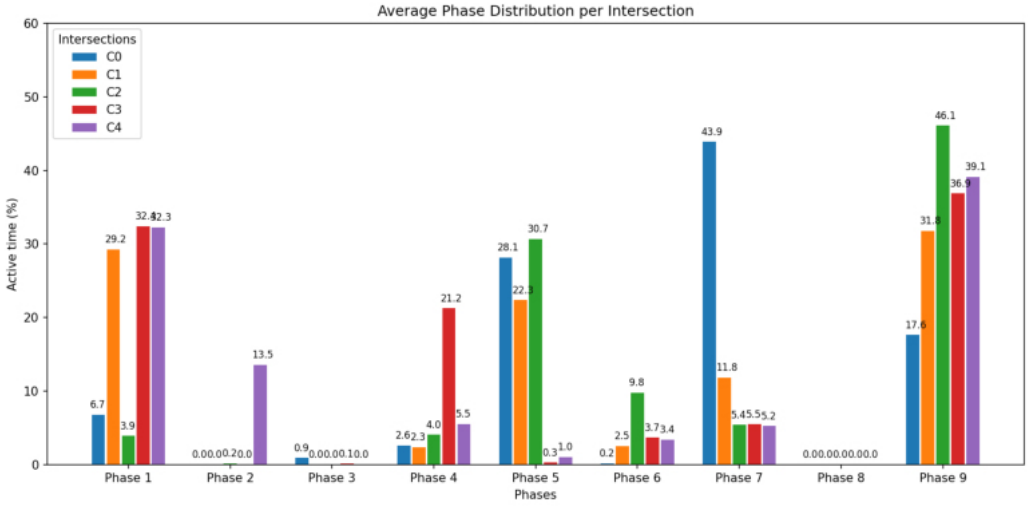
from the lower vehicle density and greater availability to serve pedestrians more frequently.



(a) *DQN*



(b) *QT-DQN*



(c) *MAPPO*

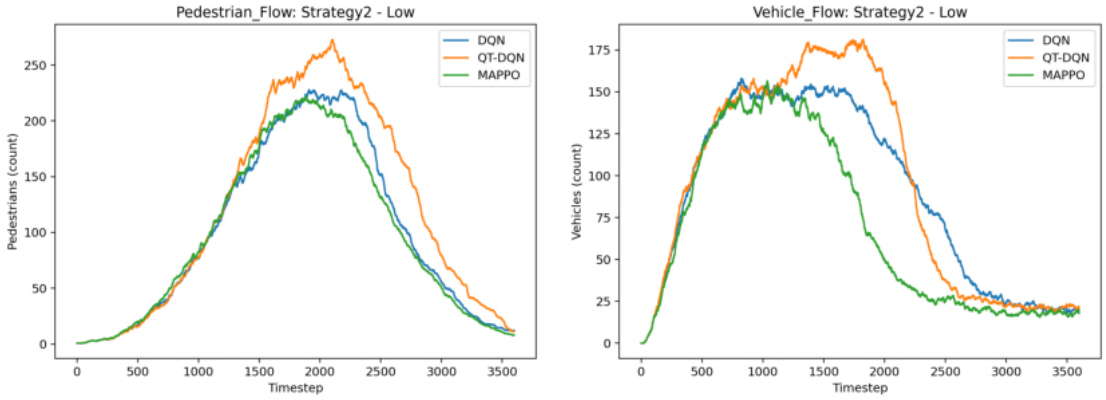
Figure 5.23: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 1, Low scenario).*

5.2.2 Strategy 2

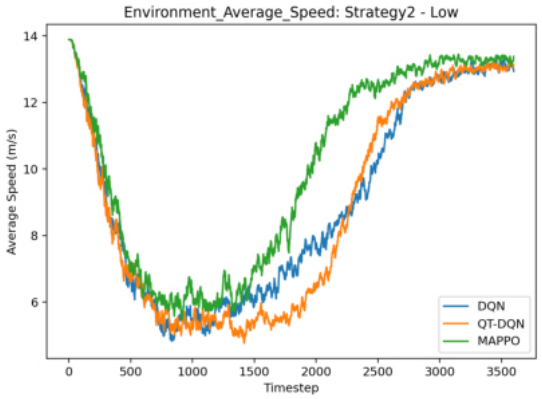
5.2.2.1 Pedestrian Curve, Vehicle Curve and Environment Average Speed

With more homogeneous flows, all methods improve relative to Strategy 1. MAPPO still clears the network faster and reaches maximum speeds earlier. QT-DQN once again performs worse for pedestrians, showing higher accumulation and less consistent clearance.

For vehicles, when compared to QT-DQN, tends to accumulate more at peak periods but resolves congestion more quickly afterward, resulting in a slightly faster recovery. Nevertheless, both DQN-based methods remain below MAPPO in final clearance and overall stability.



(a) *Pedestrian Curve* (b) *Vehicle Curve*



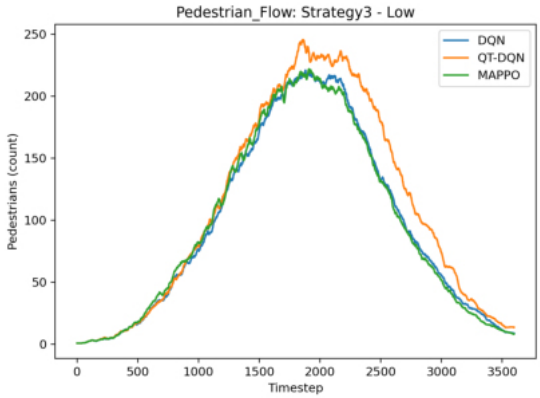
(c) *Environment Average Speed*

Figure 5.24: *Global metrics (Strategy 2, Low scenario).*

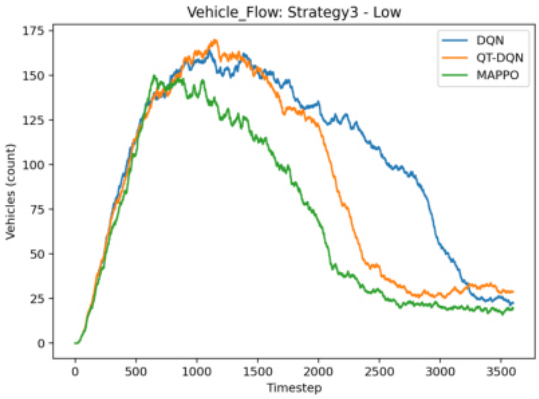
5.2.3 Strategy 3

5.2.3.1 Pedestrian Curve, Vehicle Curve and Environment Average Speed

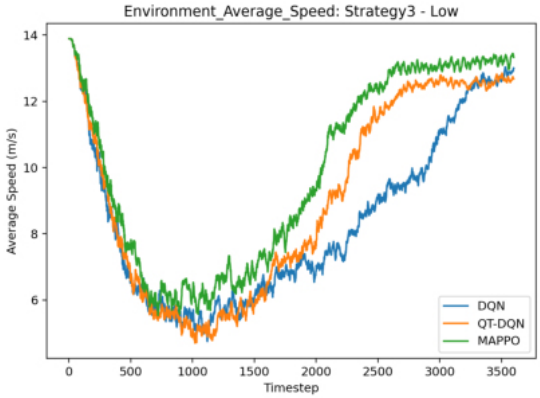
Results echo Strategy 2: MAPPO achieves the cleanest clearance and highest late-episode speeds; QT-DQN remains slightly ahead of DQN for vehicles, but behind MAPPO. Pedestrian curves are similar across methods, as expected when flows are primarily structured for vehicles.



(a) *Pedestrian Curve*



(b) *Vehicle Curve*



(c) *Environment Average Speed*

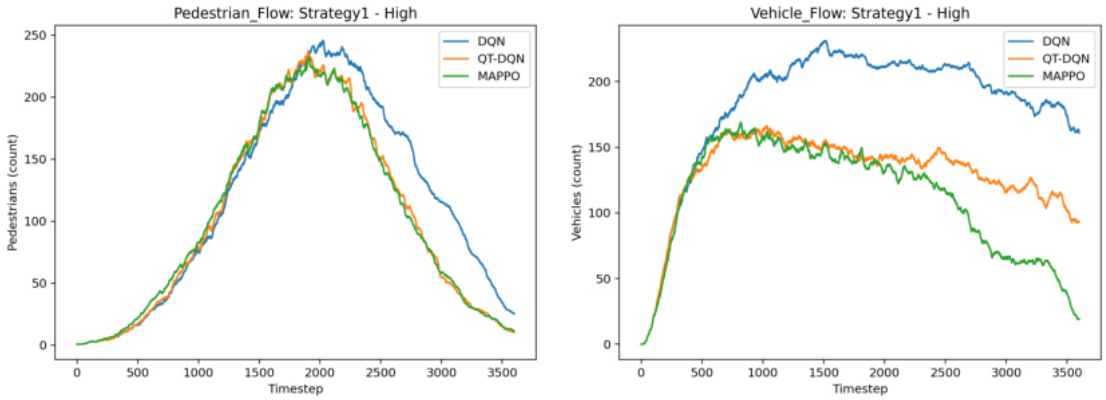
Figure 5.25: *Global metrics (Strategy 3, Low scenario).*

5.3 High-Demand Scenario

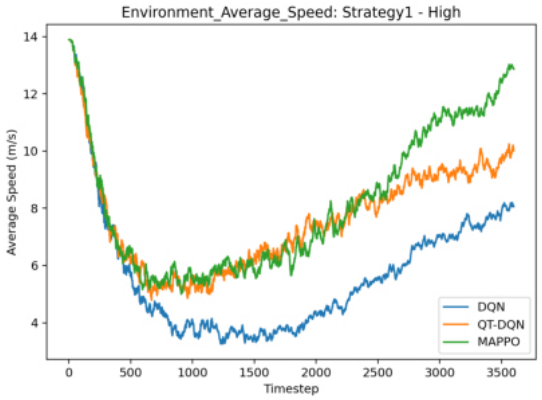
5.3.1 Strategy 1

5.3.1.1 Pedestrian Curve, Vehicle Curve and Environment Average Speed

Under high-demand conditions, all methods experience saturation effects. DQN performs noticeably worse in this regime, struggling to manage both vehicle and pedestrian flows. QT-DQN handles vehicle traffic somewhat better than DQN but still fails to achieve full stabilization. MAPPO once again distinguishes itself, clearing congestion more effectively and reaching higher environment speeds — even surpassing the DQN-based methods' performance observed under light traffic conditions.



(a) *Pedestrian Curve* (b) *Vehicle Curve*

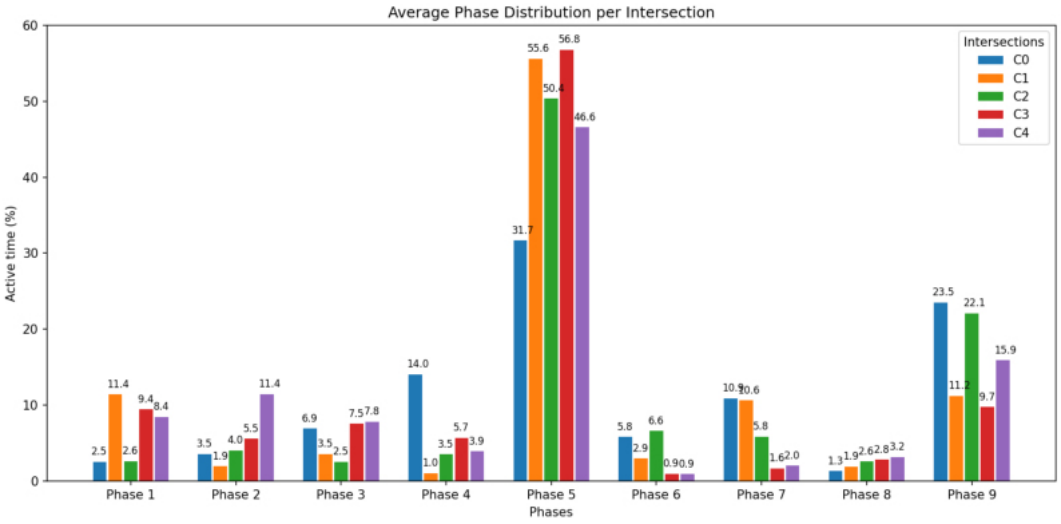


(c) *Environment Average Speed*

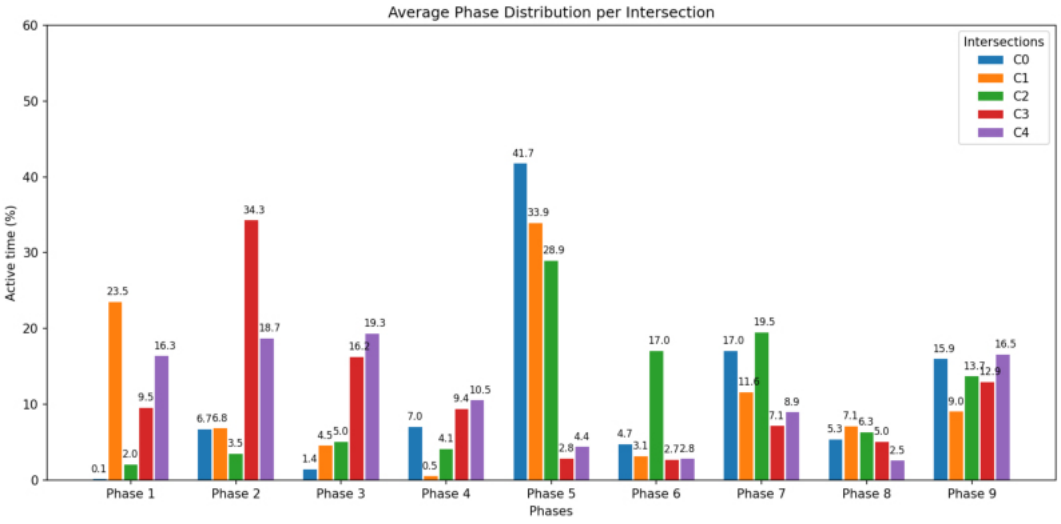
Figure 5.26: *Global metrics (Strategy 1, High scenario).*

5.3.1.2 Average Phase Distribution

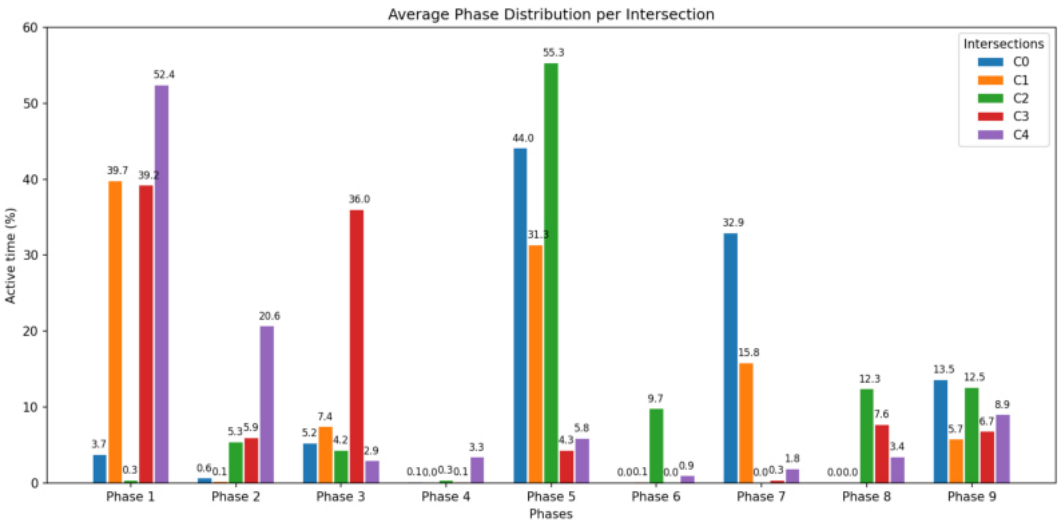
Phase usage under high demand generally follows the same trends observed in Fig. 5.7. Both QT-DQN and MAPPO exhibit the expected behavior, distributing phase activations consistently with traffic demand. In contrast, DQN shows a clear imbalance, remaining locked in phase 5 for roughly half of the time across all intersections. This excessive persistence reflects a weaker ability to adapt to dynamic conditions and contributes to the poorer overall performance observed in this scenario.



(a) *DQN*



(b) *QT-DQN*



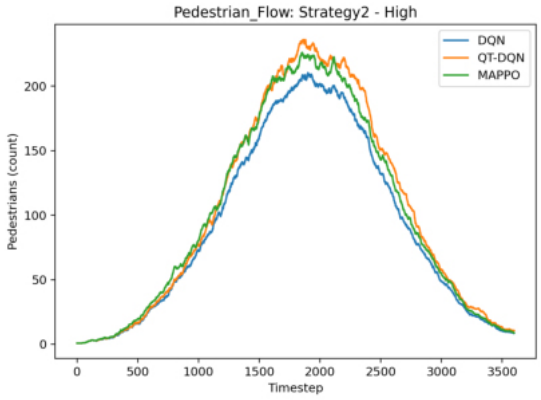
(c) *MAPPO*

Figure 5.27: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 1, High scenario).*

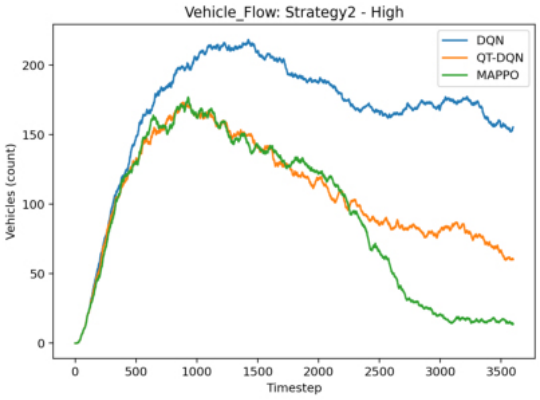
5.3.2 Strategy 2

5.3.2.1 Pedestrian Curve, Vehicle Curve and Environment Average Speed

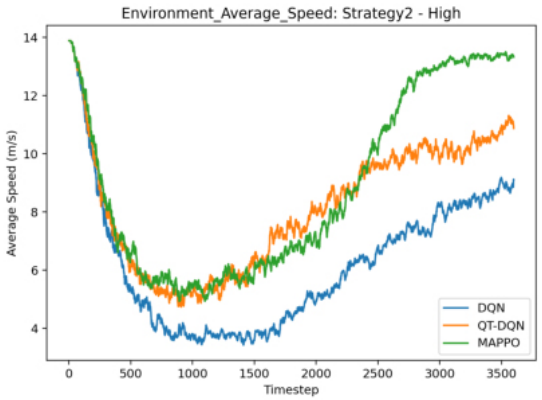
Strategy 2's homogeneity mitigates some high-demand stress. DQN, however, can become problematic especially at C1 due to excessive pedestrian-phase activations (as seen in Figure B.18), which depresses vehicle performance and fairness. QT-DQN is closer to MAPPO for much of the time but fails to clear the last vehicles. MAPPO maintains the best overall clearance and stabilizes the environment speeds earlier.



(a) *Pedestrian Curve*



(b) *Vehicle Curve*



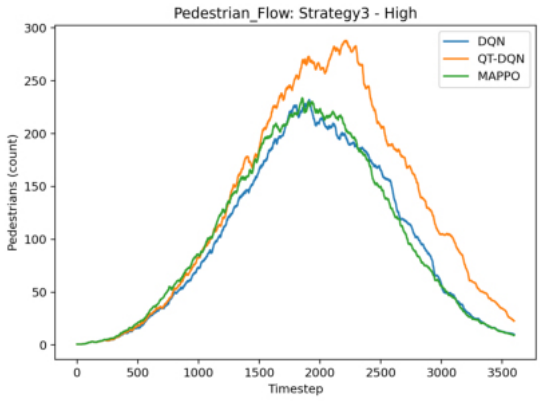
(c) *Environment Average Speed*

Figure 5.28: *Global metrics (Strategy 2, High scenario).*

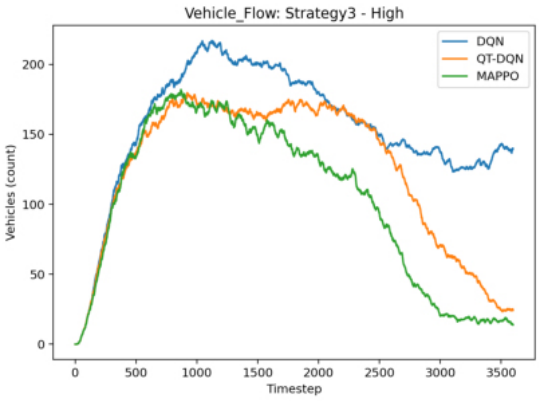
5.3.3 Strategy 3

5.3.3.1 Pedestrian Curve, Vehicle Curve and Environment Average Speed

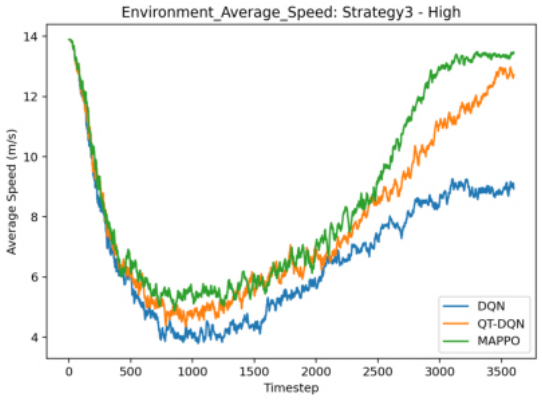
Strategy 3 follows the same general hierarchy observed in previous cases, with MAPPO outperforming both DQN-based methods. For vehicles and environment speed, MAPPO maintains the best results, though QT-DQN narrows the gap and performs better than standard DQN. Despite taking longer than MAPPO, QT-DQN manages to clear almost all vehicles by the end of the simulation period. Regarding pedestrians, QT-DQN once again performs worse, showing less consistent management of pedestrian phases, while DQN and MAPPO maintain more regular and stable pedestrian flow throughout the scenario.



(a) *Pedestrian Curve*



(b) *Vehicle Curve*



(c) *Environment Average Speed*

Figure 5.29: *Global metrics (Strategy 3, High scenario).*

Note: For the low and high demand scenarios, detailed plots of per intersection vehicle and pedestrian halting, as well as per-intersection speed results, are provided in Appendix B, together with the phase distribution analyses for Strategies 2 and 3. They corroborate the trends summarized here: MAPPO consistently maintains lower and more stable halting (vehicles and pedestrians), achieves higher per-node speeds (notably at C1), and clears the network faster. DQN and QT-DQN remain more oscillatory and less reliable, with QT-DQN typically favoring vehicle flow over pedestrian fairness relative to standard DQN.

6 Conclusion

6.1 Final Comments

The results across all evaluated metrics consistently indicate that MAPPO outperforms the other studied algorithms in the considered scenarios. Its advantage is particularly clear at intersection C1, the central node of the network, where DQN-based methods often struggled. This can be explained by theory: thanks to its centralized critic, MAPPO can evaluate the global state and coordinate agents more effectively, preventing the conflicts and omissions that occur when decisions are learned solely from local observations. QT-DQN generally performs better than DQN in vehicle-related metrics, but this improvement comes at the cost of harming pedestrian flow, while DQN itself struggles significantly under higher demand levels.

The tested strategies also influenced performance. Strategies 2 and 3, characterized by more homogeneous flows, generally produced more stable results, particularly in demanding scenarios. However, it is important to note that no strategy is universally superior: their effectiveness depends on the traffic conditions of each period, such as peak hours when people head towards their workplaces or back home.

The combination of global and local metrics provided a comprehensive view of performance. Flow and average speed curves captured the overall efficiency of each algorithm, while halting metrics revealed fairness and stability at the intersection level. These results showed that MAPPO consistently outperformed DQN-based methods, both in terms of efficiency and fairness, whereas DQN and QT-DQN displayed higher variability and less predictable behavior.

In conclusion, MAPPO emerges as the most robust and scalable approach for multi-agent traffic signal control. It ensures smoother flows, fairer treatment of vehicles and pedestrians, and superior adaptability across strategies and demand levels, making it a strong candidate for deployment in complex urban environments.

6.2 Future Work

Several directions for future research can be pursued to further extend and enhance this work. A natural next step is to study cross-strategy generalization. In particular, it remains to be studied whether training exclusively under Strategy 1 can yield robust performance when the network is operated with Strategies 2 or 3. This analysis would clarify whether it is preferable to maintain multiple strategies that reflect time-varying traffic patterns or whether MAPPO's shared policy is sufficiently expressive to cover all scenarios with a single model.

One promising direction for future research is the design, training, and evaluation of GNN architectures for traffic signal control. Comparing GNN-based methods with the algorithms studied in this work would provide a deeper understanding of their relative strengths and limitations in terms of coordination, scalability, and overall performance.

Further improvements could be achieved by introducing a wider variety of vehicle types, such as motorcycles, buses, and trucks. Differences in size, acceleration, and speed profiles might significantly influence congestion dynamics and signal control behavior, making it essential to evaluate the robustness of the proposed methods under heterogeneous vehicle characteristics.

Another valuable extension involves the inclusion of bicycles and dedicated cycling infrastructure. Although the current simulations focus primarily on light vehicles and pedestrians, integrating bicycles would enhance realism and broaden the applicability of the system. According to SUMO's documentation, bicycle simulation is developing feature and still carries certain limitations, which makes this an interesting challenge for future work.

Additionally, incorporating priority vehicles — such as ambulances, fire trucks, or public transport — would add an important dimension of practical relevance. Developing adaptive control strategies capable of dynamically adjusting signal phases to prioritize these vehicles, while minimizing disruption to general traffic, would greatly increase both the realism and societal impact of the proposed framework.

Bibliography

[1] Federal Highway Administration. *Traffic Signal Timing Manual*. Tech. rep. FHWA-HOP-08-024. U.S. Department of Transportation, Federal Highway Administration, 2008 (cit. on p. 5).

[2] R. R. Saxena. "Artificial Intelligence in Traffic Systems". In: *arXiv* (2024). DOI: 10.48550/arXiv.2412.12046 (cit. on p. 5).

[3] D. R. Shah and R. Pradhananga. "Assessment of pedestrians' red light violation behavior at signalized crosswalks in Kathmandu, Nepal". In: *Transportation Research Interdisciplinary Perspectives* 24 (2024). DOI: 10.1016/j.trip.2024.101035. URL: https://www.sciencedirect.com/science/article/pii/S2590198224000216 (cit. on p. 5).

[4] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner. "State-of-art review of traffic signal control methods: challenges and opportunities". In: *European Transport Research Review* 12.55 (2020). DOI: 10.1186/s12544-020-00439-1 (cit. on p. 6).

[5] S. Jafari, Z. Shahbazi, and Y.-C. Byun. "Improving the Performance of Single-Intersection Urban Traffic Networks Based on a Model Predictive Controller". In: *Sustainability* 13.10 (2021), p. 5630. DOI: 10.3390/su13105630. URL: https://doi.org/10.3390/su13105630 (cit. on p. 6).

[6] D. Tang and Y. Duan. "Traffic Signal Control Optimization Based on Neural Network in the Framework of Model Predictive Control". In: *Actuators* 13.7 (2024), p. 251. DOI: 10.3390/act13070251 (cit. on p. 6).

[7] G. Galvão, M. Vieira, M. A. Vieira, M. Véstias, and P. Louro. "Integration of Visible Light Communication and Deep Reinforcement Learning to Enhance Urban Traffic Management". In: *2025 9th International Young Engineers Forum on Electrical and Computer Engineering (YEF-ECE)*. IEEE, 2025. DOI: 10.1109/YEF-ECE61178.2025.11117529. URL: https://ieeexplore.ieee.org/document/11117529 (cit. on p. 6).

[8] M. Muresan, L. Fu, and G. Pan. *Adaptive Traffic Signal Control with Deep Reinforcement Learning: An Exploratory Investigation*. 2019. DOI: 10.48550/arXiv.1901.00960 (cit. on p. 6).

[9] M. A. Vieira, G. Galvão, M. Vieira, T. Antunes, M. Véstias, and P. Louro. "Decentralized Multi-Agent Reinforcement Learning with Visible Light Communication for Robust Urban Traffic Signal Control". Manuscript submitted for publication in Sensors. 2025 (cit. on p. 7).

[10] P. Michailidis, I. Michailidis, C. R. Lazaridis, and E. Kosmatopoulos. "Traffic Signal Control via Reinforcement Learning: A Review on Applications and Innovations". In: *Infrastructures* 10.5 (2025). DOI: 10.3390/infrastructures10050114 (cit. on p. 7).

[11] S. K. Yang, J. C. Li, and H. B. Shi. "Mix-attention approximation for homogeneous large-scale multi-agent reinforcement learning". In: *Neural Computing & Applications* 35.4 (2023), pp. 3143–3154. DOI: 10.1007/s00521-022-07880-4 (cit. on p. 7).

[12] C. Zhu, M. Dastani, and S. Wang. "A survey of multi-agent deep reinforcement learning with communication". In: *Autonomous Agents and Multi-Agent Systems* 38.4 (2024). DOI: 10.1007/s10458-023-09633-6 (cit. on p. 7).

[13] C. Amato. "An Initial Introduction to Cooperative Multi-Agent Reinforcement Learning". In: *arXiv* (2024). URL: https://arxiv.org/abs/2405.06161 (cit. on p. 8).

[14] A. Han, J. Hu, P. Wei, Z. Zhang, Y. Guo, J. Lu, and Z. Zhang. "JoyAgents-R1: Joint Evolution Dynamics for Versatile Multi-LLM Agents with Reinforcement Learning". In: *arXiv* (2025). URL: https://arxiv.org/abs/2506.19846 (cit. on p. 8).

[15] L. Huang and X. Qu. "Improving traffic signal control operations using proximal policy optimization". In: *IET Intelligent Transport Systems* 17.1 (2022). DOI: 10.1049/itr2.12286 (cit. on p. 8).

[16] S. Hu, M. A. Hady, J. Qiao, J. Cao, M. Pratama, and R. Kowalczyk. "Adaptability in Multi-Agent Reinforcement Learning: A Framework and Unified Review". In: *arXiv* (2025). DOI: 10.48550/arXiv.2507.10142 (cit. on p. 8).

[17] G. Galvão, M. A. Vieira, M. Vieira, M. Véstias, P. Louro, and R. Jardim-Gonçalves. "Innovative integration of visible light communication and artificial intelligence to enhance urban traffic management". In: *AI and Optical Data Sciences VI*. Proceedings of SPIE. Vol. 13375. SPIE, 2025. DOI: 10.1117/12.3039137 (cit. on p. 9).

[18] T. Antunes, G. Galvão, M. A. Vieira, M. Vieira, M. Véstias, and P. Louro. "Intelligent Intersection Management through Multi-Agent Reinforcement Learning and Visible Light Communication Integration". In: *Proceedings of the 11th International Conference on Sensors and Electronic Instrumentation Advances (SEIA' 2025)*. 2025 (cit. on p. 10).

[19] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan. "Cooperative Deep Q-Learning with Q-Value Transfer for Multi-Intersection Signal Control". In: *IEEE Access* 7 (2019), pp. 40797–40809. DOI: 10.1109/ACCESS.2019.2907618 (cit. on pp. 10, 30).

[20] G. Galvão, M. A. Vieira, M. Vieira, M. Véstias, P. Louro, and R. Jardim-Gonçalves. "Integrating Visible Light Communication and AI for Adaptive Traffic Management: A Focus on Reward Functions and Rerouting Coordination". In: *Applied Sciences* 15.1 (2025), p. 116. DOI: 10.3390/app15010116 (cit. on p. 10).

[21] Y. He, Y. H. Wang, F. R. Yu, Q. Z. Lin, J. Q. Li, and V. C. M. Leung. "Efficient Resource Allocation for Multi-Beam Satellite-Terrestrial Vehicular Networks: A Multi-Agent Actor Critic Method With Attention Mechanism". In: *IEEE Transactions on Intelligent Transportation Systems* 23.3 (2022), pp. 2727–2738. DOI: 10.1109/TITS.2021.3128209 (cit. on p. 10).

[22] S. Choi, D.-Y. Yeung, and N. Zhang. "An environment model for nonstationary reinforcement learning". In: *Advances in Neural Information Processing Systems (NeurIPS 1999)*. Vol. 12. MIT Press, 1999 (cit. on p. 10).

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal Policy Optimization Algorithms". In: *arXiv preprint arXiv:1707.06347* (2017) (cit. on p. 10).

[24] N.-C. Huang, P.-C. Hsieh, K.-H. Ho, and I.-C. Wu. "PPO-Clip Attains Global Optimality: Towards Deeper Understandings of Clipping". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.11 (2024), pp. 12600–12607. DOI: 10.1609/aaai.v38i11.29154 (cit. on p. 10).

[25] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. M. Bayen, and Y. Wu. "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS 2022)*. Curran Associates Inc., 2022 (cit. on pp. 10, 11).

[26] J. Xu, Z. Zhang, S. Zhang, and J. Miao. "An Improved Traffic Signal Control Method Based on Multi-agent Reinforcement Learning". In: *2021 40th Chinese Control Conference (CCC)*. IEEE, 2021. DOI: 10.23919/CCC52363.2021.9549970 (cit. on p. 11).

[27] A. Goeckner, Y. Sui, N. Martinet, X. Li, and Q. Zhu. "Graph Neural Network-based Multi-agent Reinforcement Learning for Resilient Distributed Coordination of Multi-Robot Systems". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024. DOI: 10.1109/IROS58592.2024.10802510 (cit. on p. 11).

[28] S. Rahmani, A. Baghbani, N. Bouguila, and Z. Patterson. "Graph Neural Networks for Intelligent Transportation Systems: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 24.8 (2023), pp. 8846–8885. DOI: 10.1109/TITS.2023.3257759 (cit. on p. 12).

[29] J. M. Bilal and D. Jacob. "Intelligent Traffic Control System". In: *2007 IEEE International Conference on Signal Processing and Communications (ICSPC)*. Dubai, United Arab Emirates, 2007, pp. 496–499. DOI: 10.1109/ICSPC.2007.4728364 (cit. on p. 14).

[30] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy. "Analytical Model for Connectivity in Vehicular Ad Hoc Networks". In: *IEEE Transactions on Vehicular Technology* 57.6 (2008), pp. 3341–3356. DOI: 10.1109/TVT.2007.912321 (cit. on p. 14).

[31] W.-H. Shen and H.-M. Tsai. "Testing vehicle-to-vehicle visible light communications in real-world driving scenarios". In: *2017 IEEE Vehicular Networking Conference (VNC)*. 2017, pp. 187–194. DOI: 10.1109/VNC.2017.8275602 (cit. on p. 14).

[32] X. Liang, X. Du, G. Wang, and Z. Han. "A Deep Reinforcement Learning Network for Traffic Light Cycle Control". In: *IEEE Transactions on Vehicular Technology* 68.2 (2019), pp. 1243–1253. DOI: 10.1109/TVT.2018.2890726 (cit. on p. 14).

[33] P. Álvarez López, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner. "Microscopic Traffic Simulation using SUMO". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018. DOI: 10.1109/ITSC.2018.8569938 (cit. on p. 14).

[34] M. A. Vieira, M. Vieira, P. Louro, P. Vieira, and R. Fernandes. "Adaptive Traffic Control Using Cooperative Communication Through Visible Light". In: *SN Computer Science* 5.159 (2024). DOI: 10.1007/s42979-023-02483-9 (cit. on p. 14).

[35] R. Fernandes, M. A. Vieira, M. Vieira, P. Vieira, P. Louro, and M. Véstias. "Using visible light communication to implement intelligent traffic signals and cooperative trajectories at urban intersections". In: *Light-Emitting Devices, Materials, and Applications XXVII*. Ed. by J. K. Kim, M. R. Krames, and M. Strassburg. SPIE, 2023. DOI: 10.1117/12.2647862 (cit. on p. 15).

[36] M. A. Vieira, M. Vieira, P. Vieira, R. Fernandes, and P. Louro. "Dynamic vehicular visible light communication for traffic management". In: *Next-Generation Optical Communication: Components, Sub-Systems, and Systems XII*. Ed. by G. Li, K. Nakajima, and A. K. Srivastava. SPIE, 2023, 124290O. DOI: 10.1117/12.2647866 (cit. on p. 15).

[37] M. Vieira et al. "Vehicular Visible Light Communication for Intersection Management". In: *Signals* 4.2 (2023), pp. 457–477. DOI: 10.3390/signals4020024. URL: https://doi.org/10.3390/signals4020024 (cit. on p. 15).

[38] A. Yousefpour et al. "All one needs to know about fog computing and related edge computing paradigms: A complete survey". In: *Journal of Systems Architecture* 98 (2019), pp. 289–330 (cit. on p. 15).

[39] W. G. Najm, J. Koopmann, J. D. Smith, and J. Brewer. *Frequency of Target Crashes for IntelliDrive Safety Systems*. Tech. rep. U.S. Department of Transportation, Research and Innovative Technology Administration (RITA); National Highway Traffic Safety Administration (NHTSA), 2010 (cit. on p. 15).

A Swap Actions Function

To ensure that the Q-values obtained from different neighbors are comparable, a normalization step was introduced through the `swap_actions` function. As discussed in Section 4.2, the same action (e.g., *East–West*) may represent different traffic dynamics depending on the orientation of the neighbor. This function reorders the Q-values of each neighbor so that all actions correspond to the same semantic meaning across orientations before aggregation.

```
1
2  def _swap_actions(self, q_vals, direcao):
3      q_vals = q_vals.copy()
4      swap_pairs = []
5      if direcao == 'S':
6          swap_pairs = [(1, 2), (5, 6)]  # Symmetric actions
7      elif direcao == 'E':
8          swap_pairs = [(0, 4), (3, 7), (1, 6), (2, 6), (2, 5)]
9      elif direcao == 'W':
10         swap_pairs = [(0, 4), (3, 7), (1, 5), (2, 5), (2, 6)]
11     for a, b in swap_pairs:
12         q_vals[a], q_vals[b] = q_vals[b], q_vals[a]
13     return q_vals
```

Listing A.1: Implementation of the `swap_actions` function.

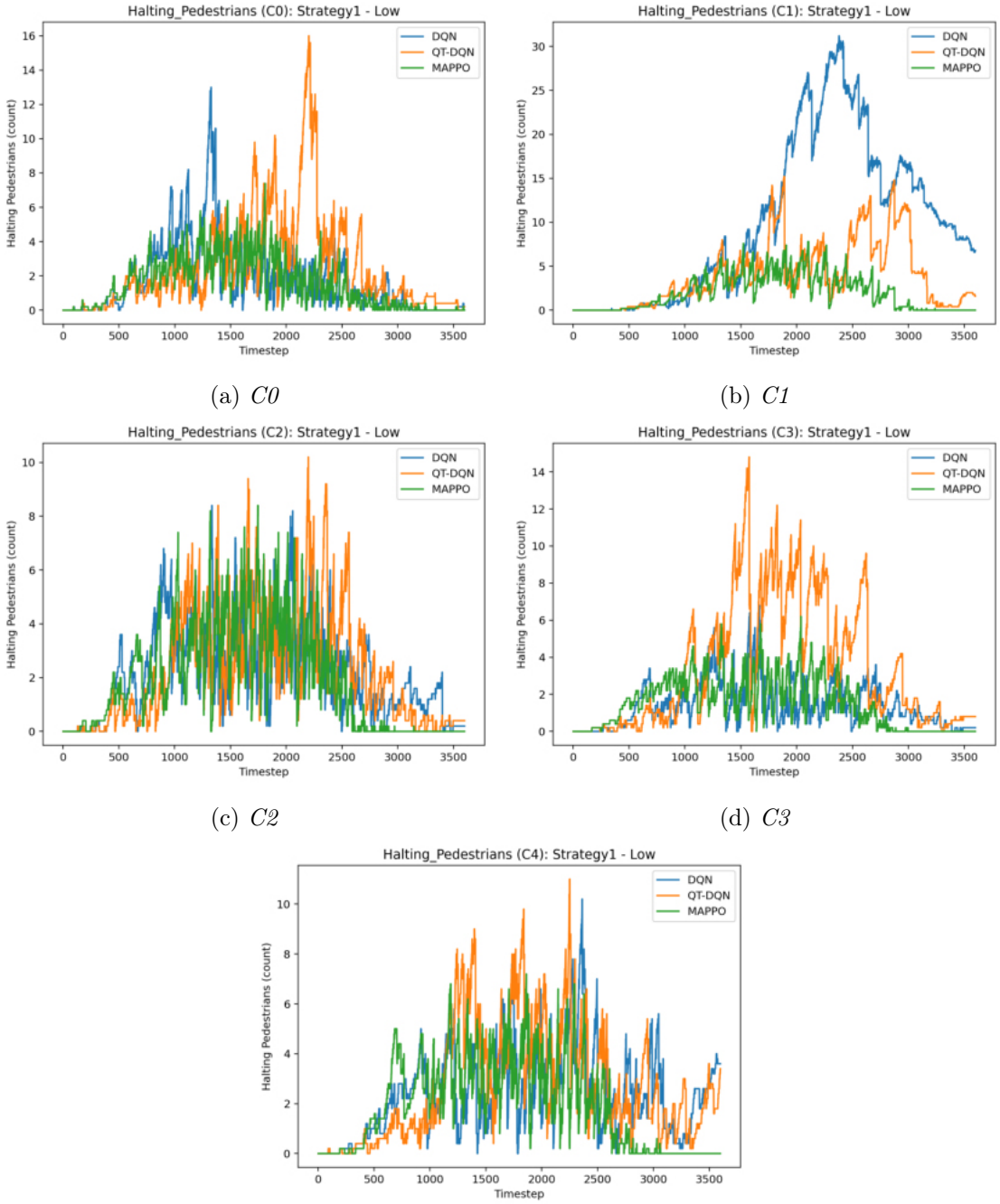
This procedure guarantees that the aggregated Q-values from different neighbors are aligned and consistent, allowing cooperative information to be exploited effectively during learning.

B Halting and Speed metrics for Low- and High-Demand Scenarios

B.1 Low-Demand Scenario

B.1.1 Strategy 1

B.1.1.1 Pedestrian Halting



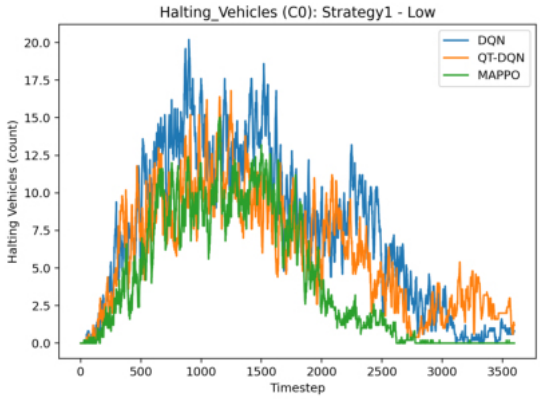
(a) *C0* (b) *C1*

(c) *C2* (d) *C3*

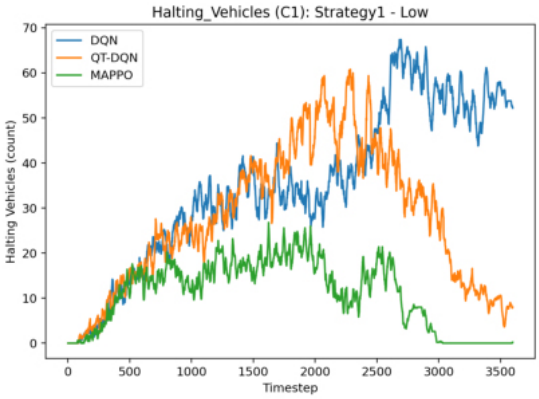
(e) *C4*

Figure B.1: *Halting pedestrians per intersection (Low-demand) — Strategy 1.*

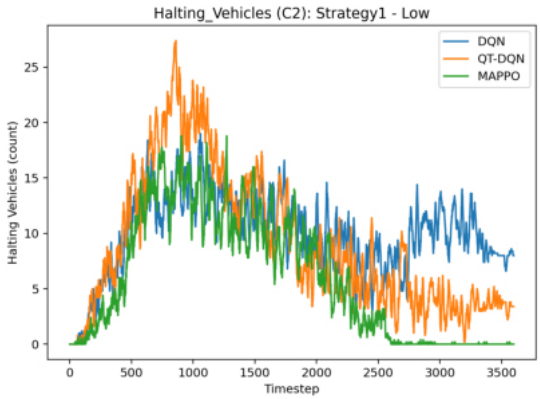
B.1.1.2 Vehicle Halting



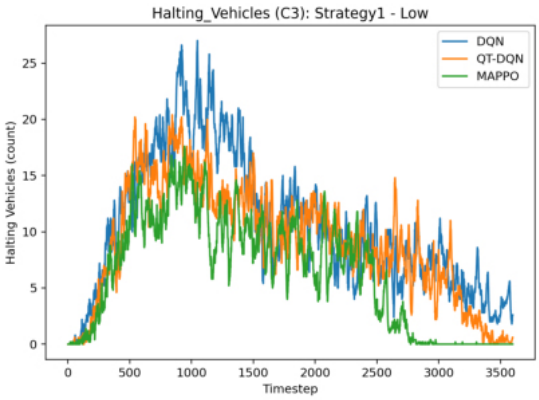
(a) *C0*



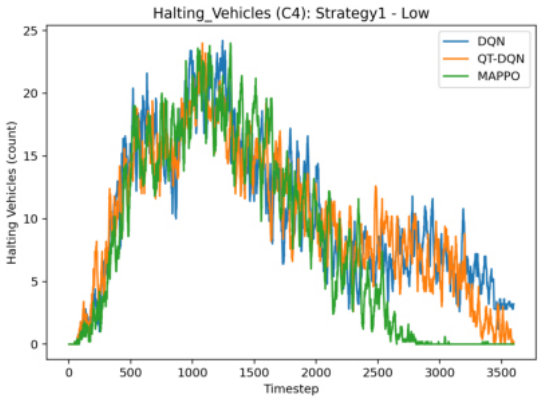
(b) *C1*



(c) *C2*



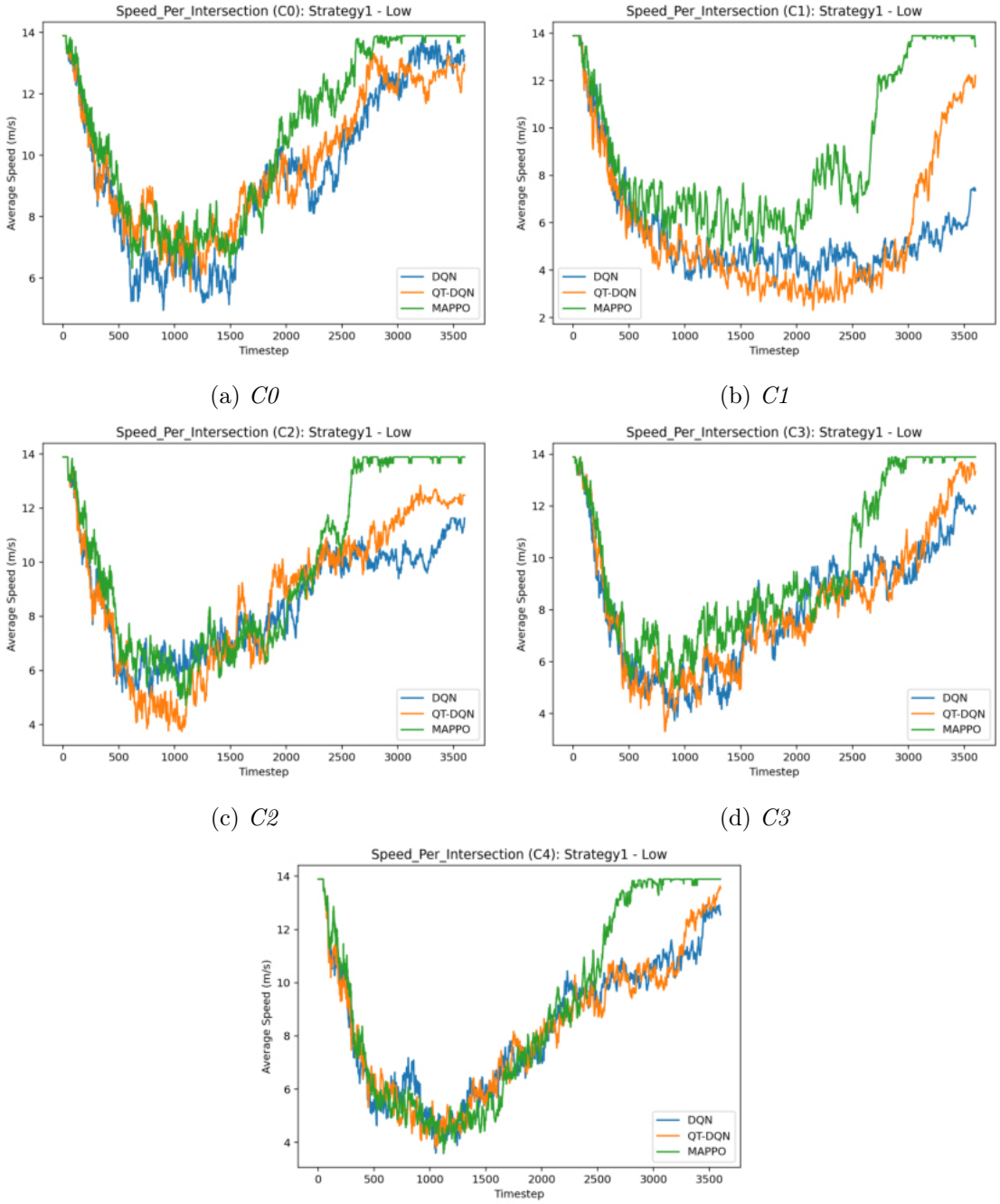
(d) *C3*



(e) *C4*

Figure B.2: *Halting vehicles per intersection (Low-demand) — Strategy 1.*

B.1.1.3 Average Speed per Intersection



(a) *C0*

(b) *C1*

(c) *C2*

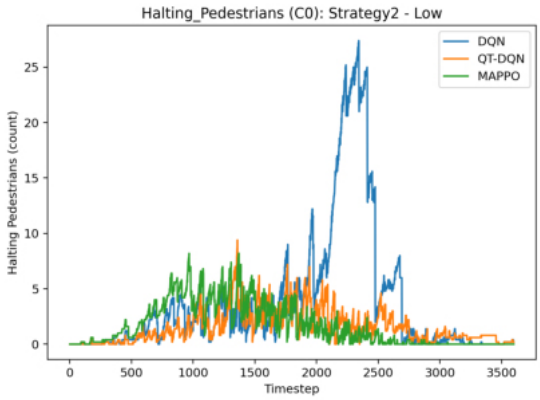
(d) *C3*

(e) *C4*

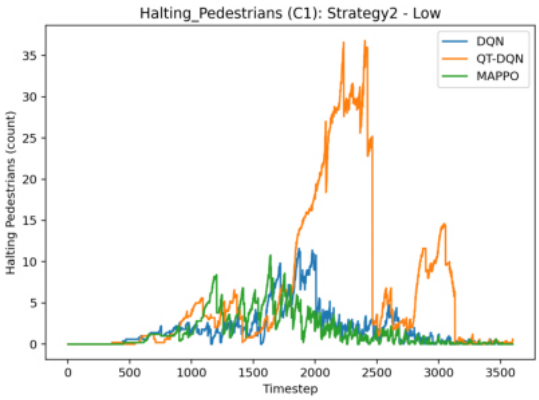
Figure B.3: *Average speed per intersection (Low-demand) — Strategy 1.*

B.1.2 Strategy 2

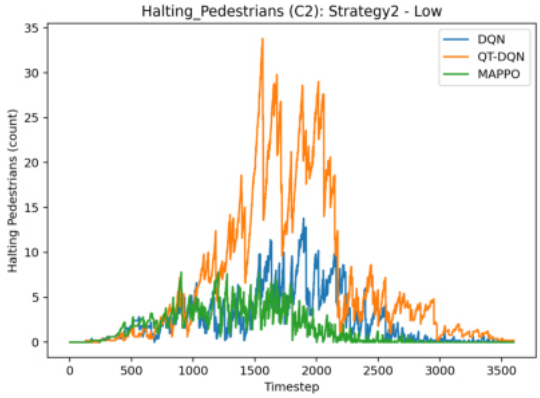
B.1.2.1 Pedestrian Halting



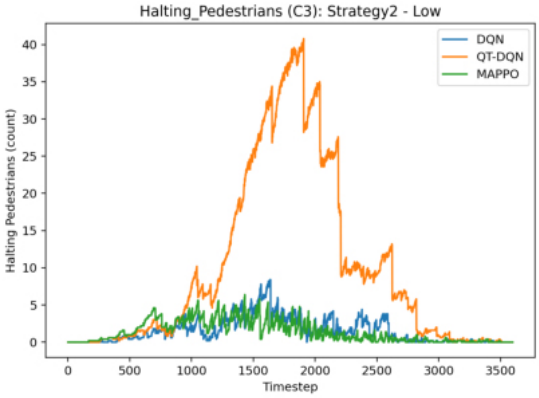
(a) *C0*



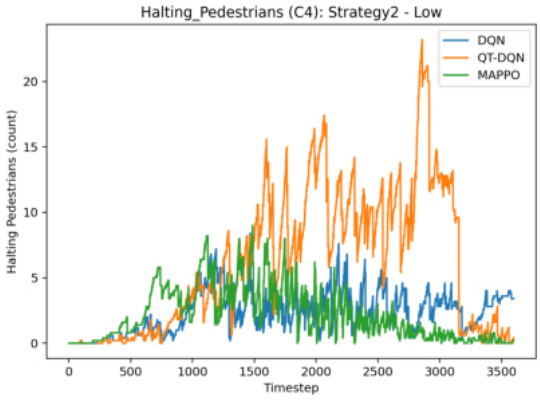
(b) *C1*



(c) *C2*



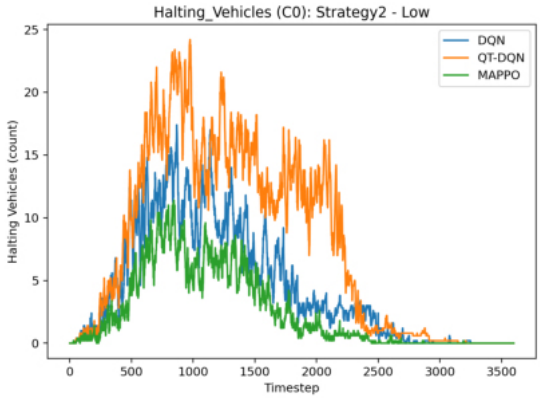
(d) *C3*



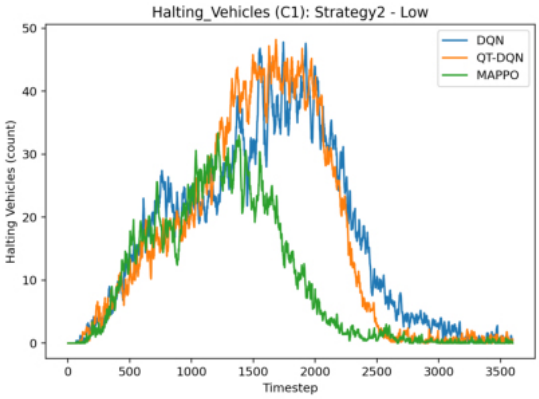
(e) *C4*

Figure B.4: *Halting pedestrians per intersection (Low-demand) — Strategy 2.*

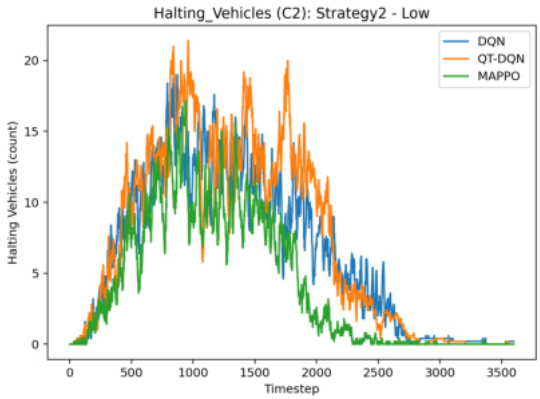
B.1.2.2 Vehicle Halting



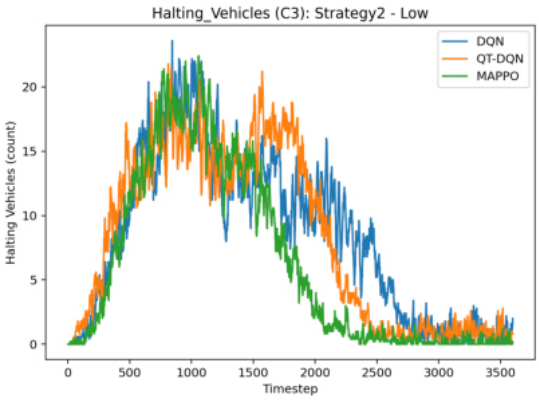
(a) *C0*



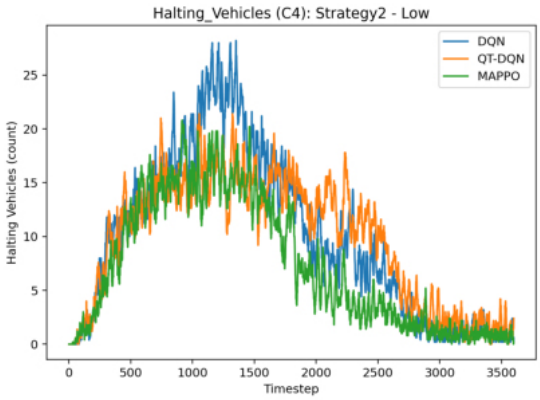
(b) *C1*



(c) *C2*



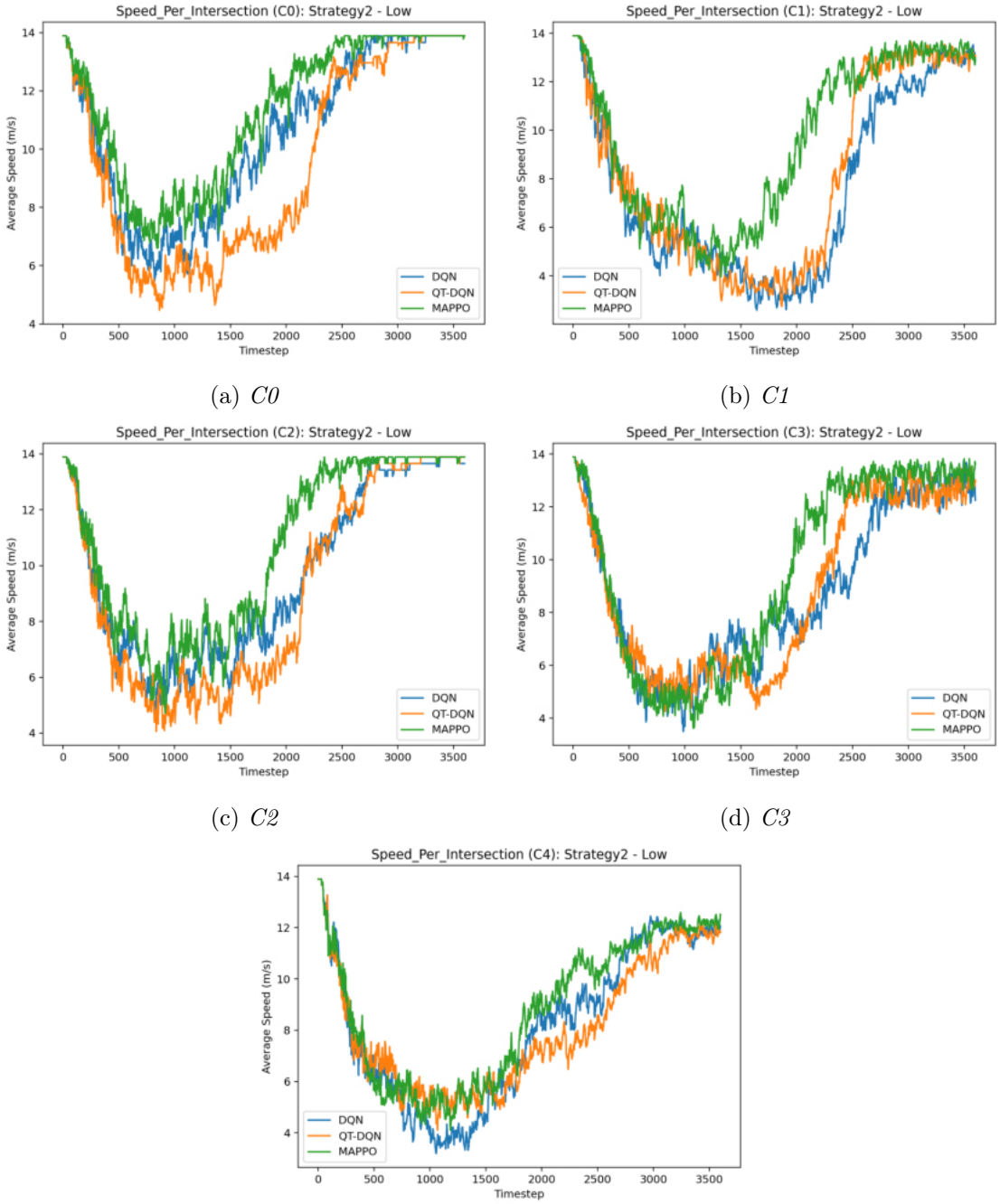
(d) *C3*



(e) *C4*

Figure B.5: *Halting vehicles per intersection (Low-demand) — Strategy 2.*

B.1.2.3 Average Speed per Intersection



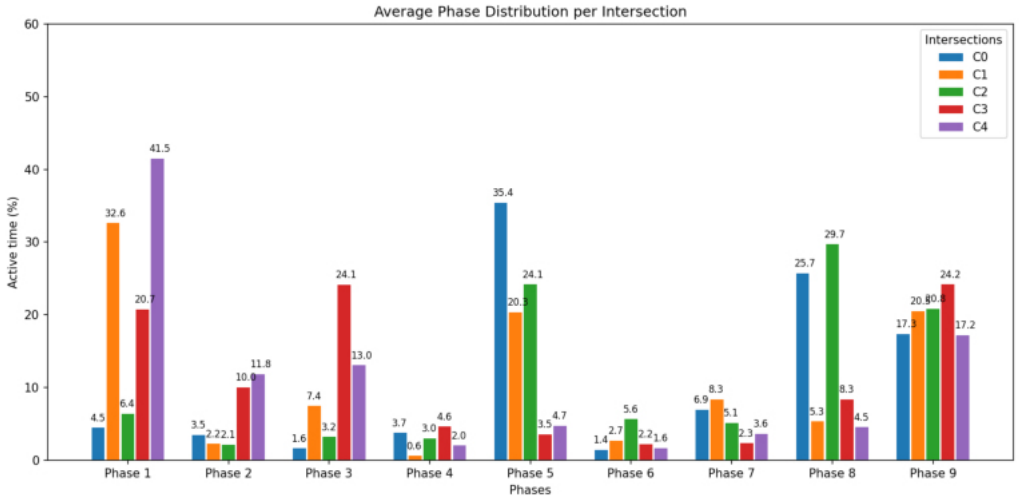
(a) *C0* (b) *C1*

(c) *C2* (d) *C3*

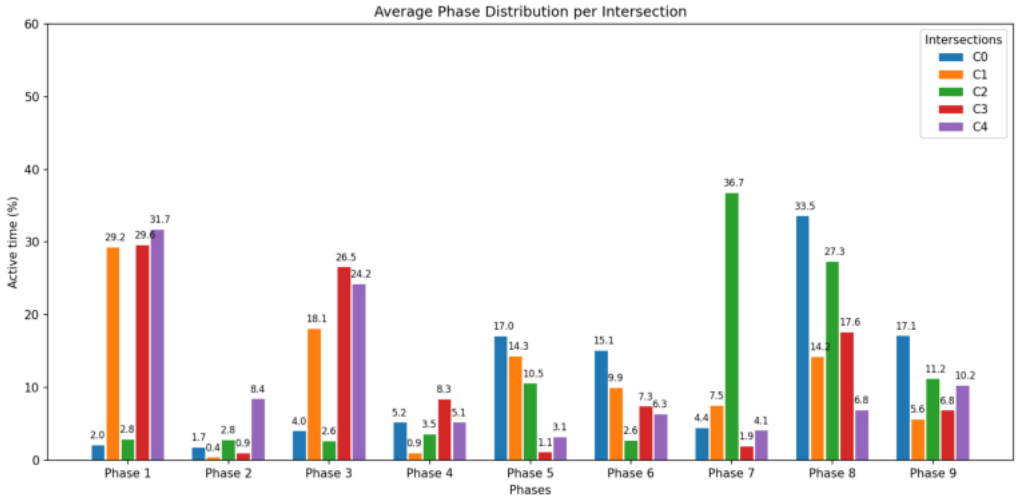
(e) *C4*

Figure B.6: *Average speed per intersection (Low-demand) — Strategy 2.*

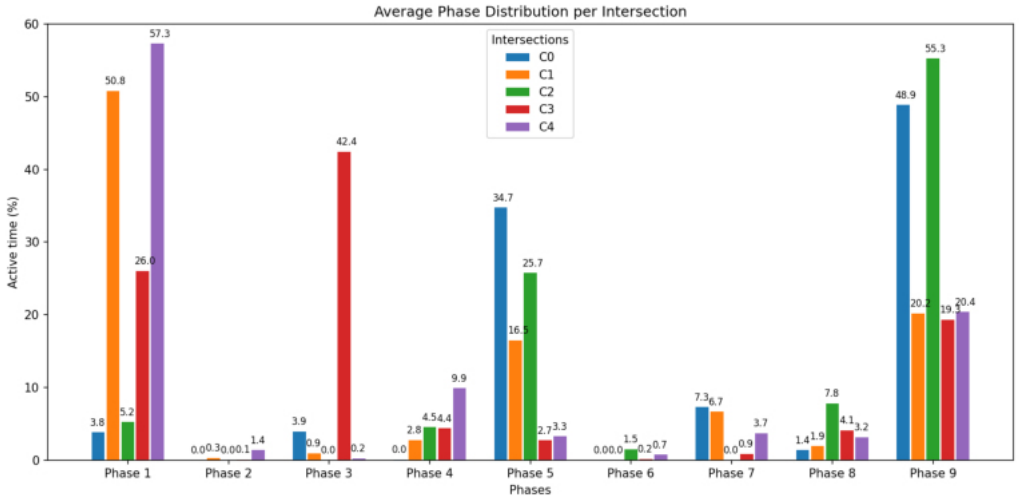
B.1.2.4 Average Phase Distribution



(a) *DQN*



(b) *QT-DQN*

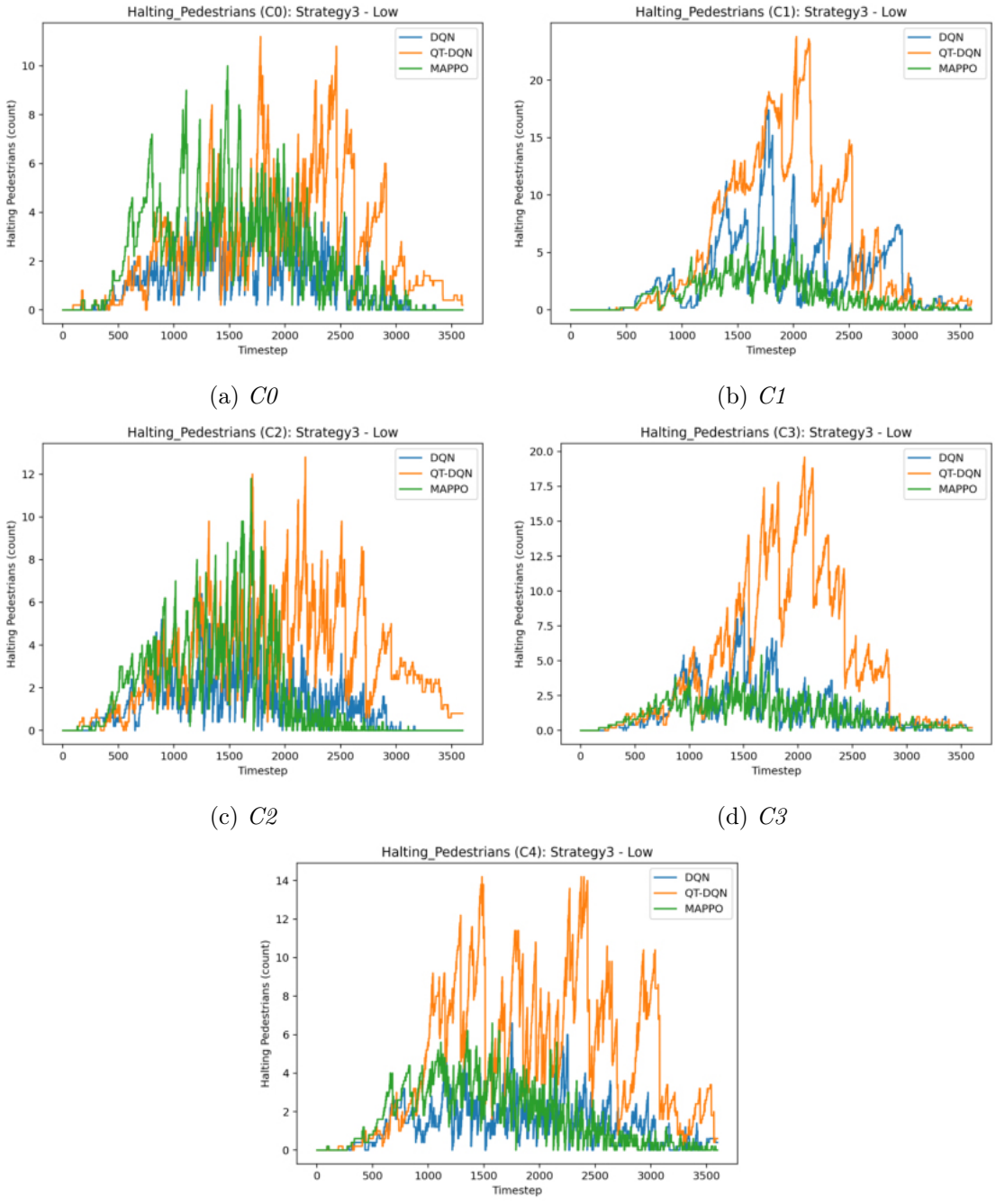


(c) *MAPPO*

Figure B.7: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 2, Low scenario).*

B.1.3 Strategy 3

B.1.3.1 Pedestrian Halting



(a) *C0*

(b) *C1*

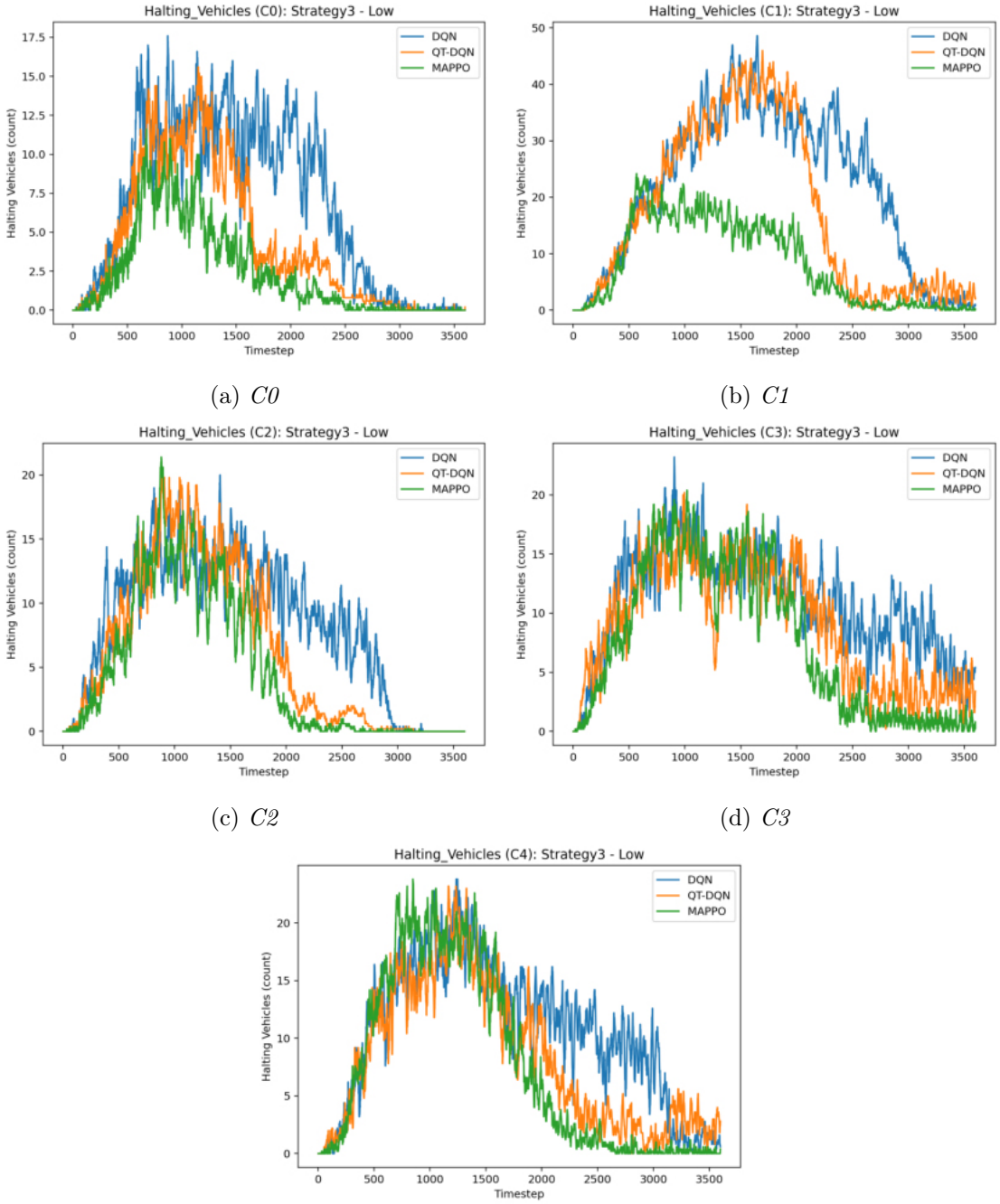
(c) *C2*

(d) *C3*

(e) *C4*

Figure B.8: *Halting pedestrians per intersection (Low-demand) — Strategy 3.*

B.1.3.2 Vehicle Halting



(a) *C0*

(b) *C1*

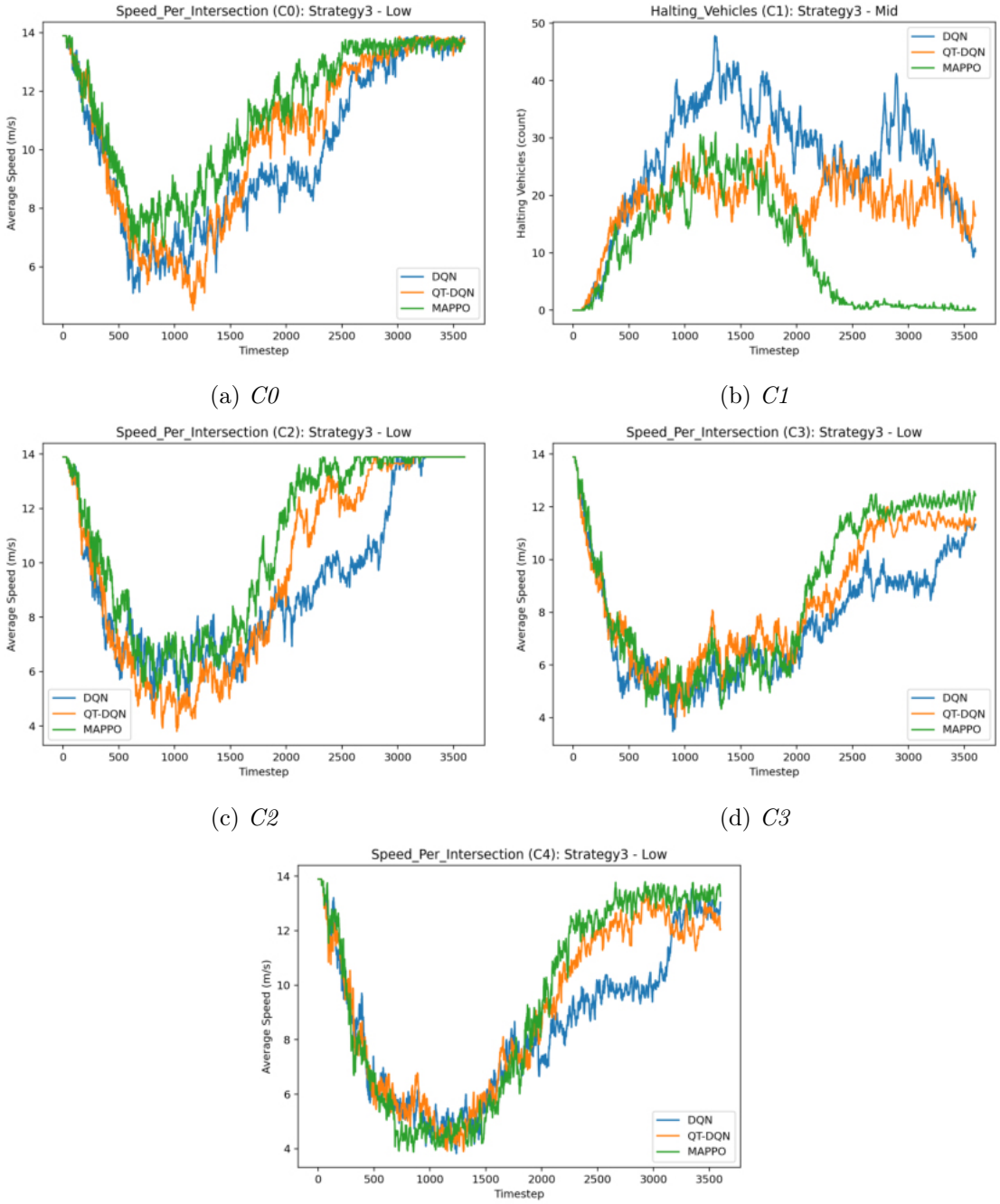
(c) *C2*

(d) *C3*

(e) *C4*

Figure B.9: *Halting vehicles per intersection (Low-demand) — Strategy 3.*

B.1.3.3 Average Speed per Intersection



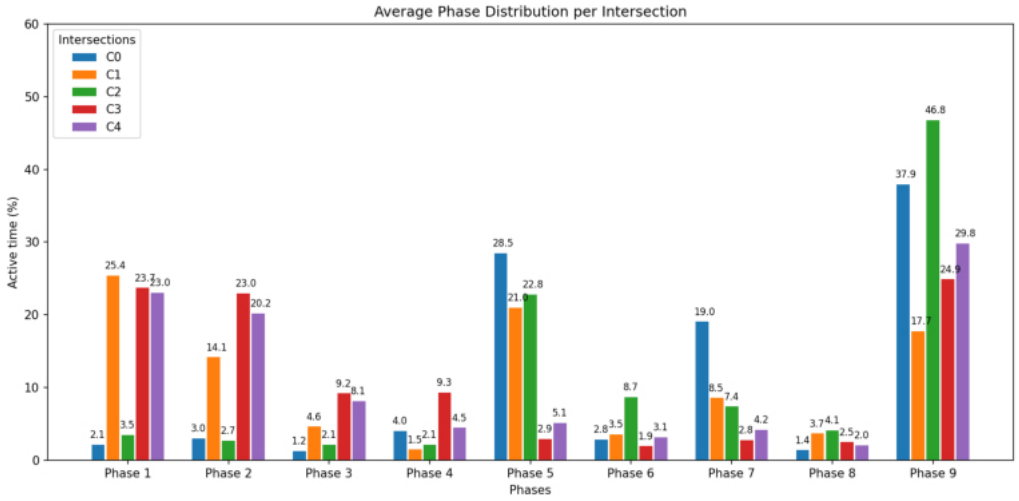
(a) *C0* (b) *C1*

(c) *C2* (d) *C3*

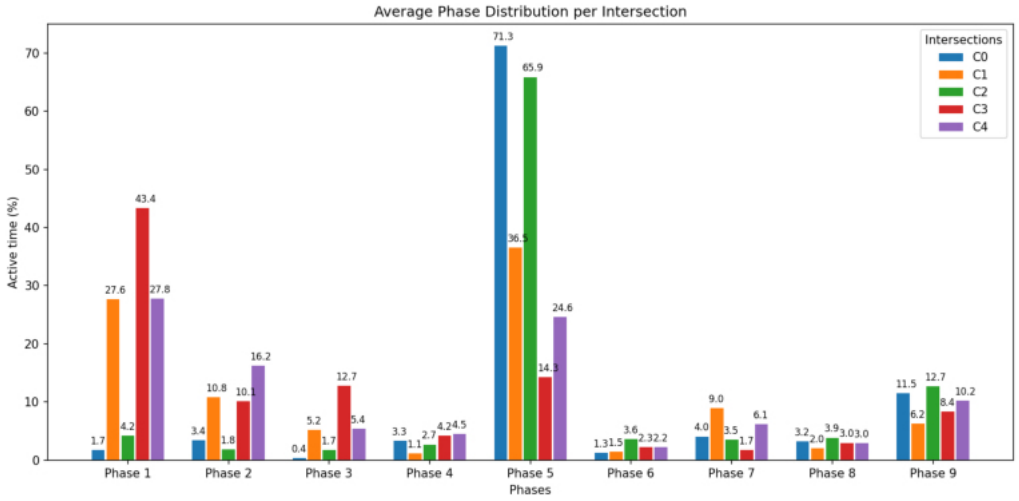
(e) *C4*

Figure B.10: *Average speed per intersection (Low-demand) — Strategy 3.*

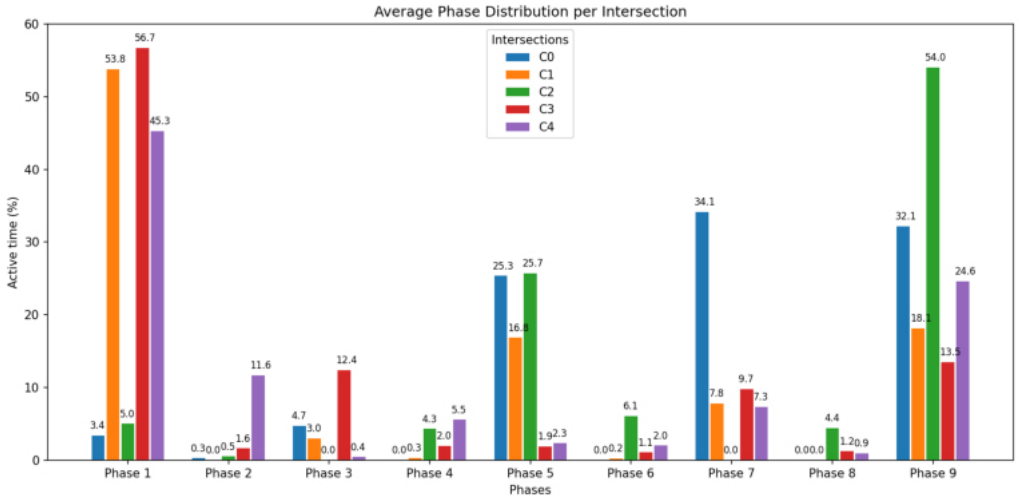
B.1.3.4 Average Phase Distribution



(a) *DQN*



(b) *QT-DQN*



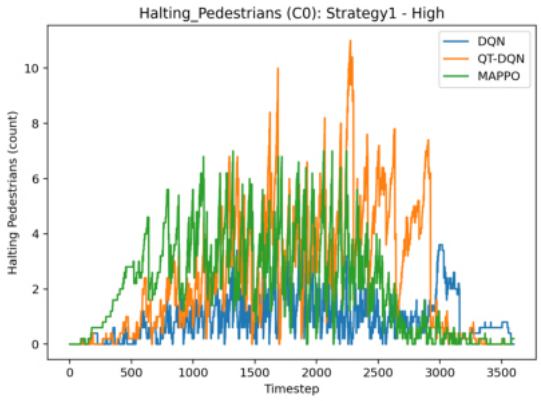
(c) *MAPPO*

Figure B.11: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 3, Low scenario).*

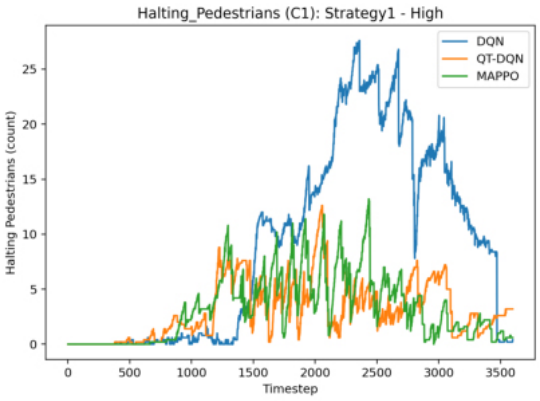
B.2 High-Demand Scenario

B.2.1 Strategy 1

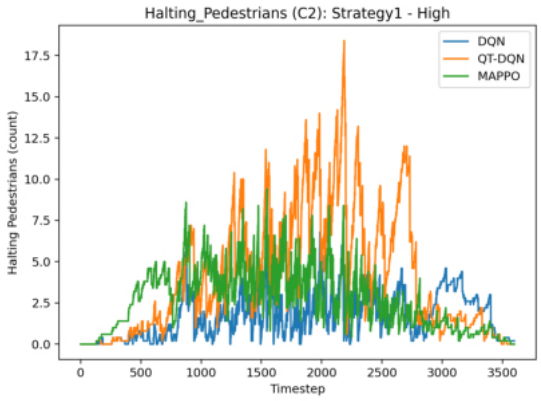
B.2.1.1 Pedestrian Halting



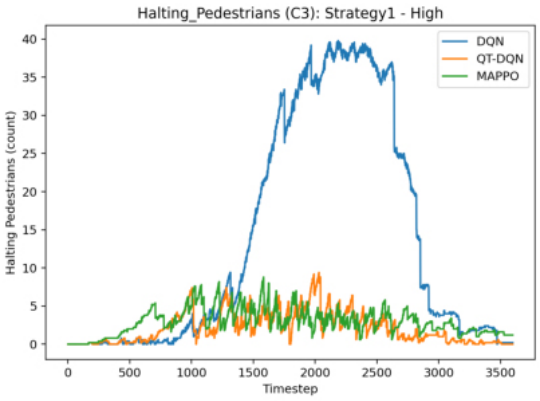
(a) *C0*



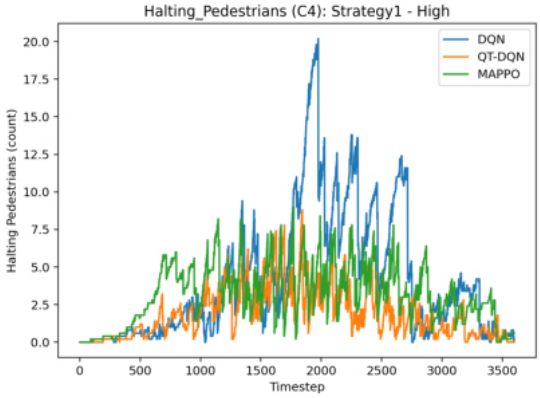
(b) *C1*



(c) *C2*



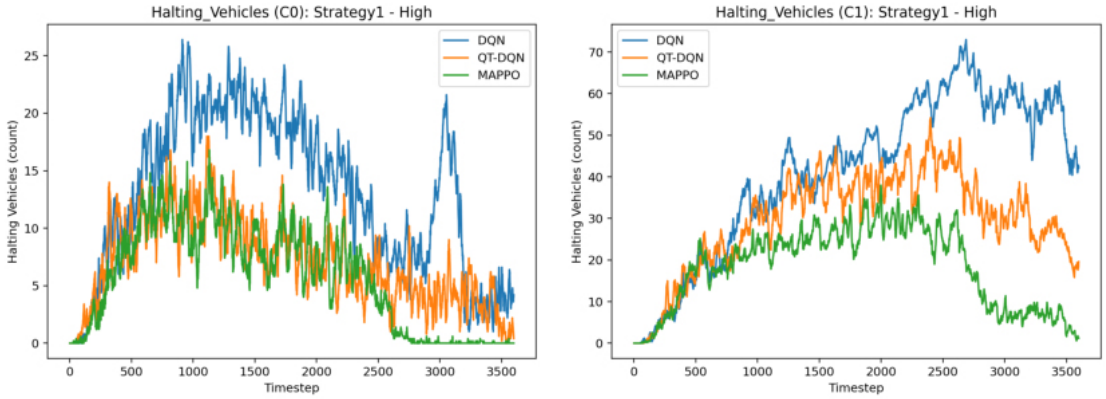
(d) *C3*



(e) *C4*

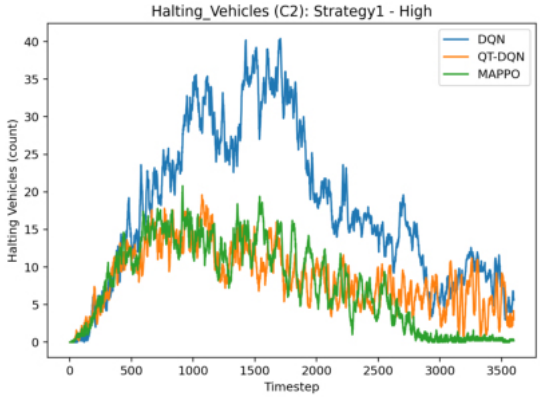
Figure B.12: *Halting pedestrians per intersection (High-demand) — Strategy 1.*

B.2.1.2 Vehicle Halting

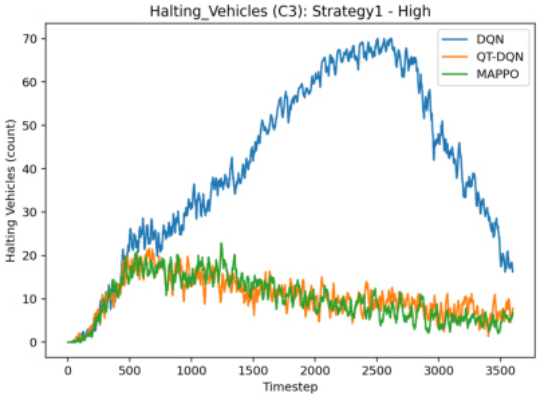


(a) *C0*

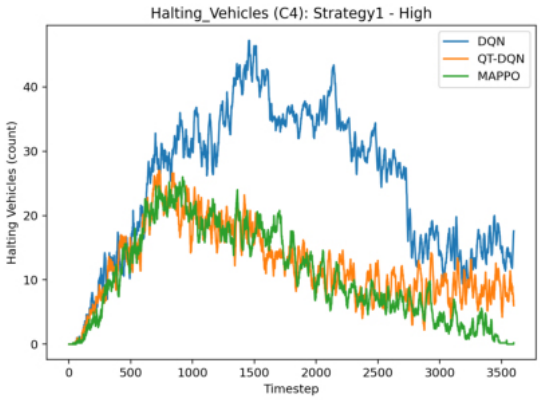
(b) *C1*



(c) *C2*



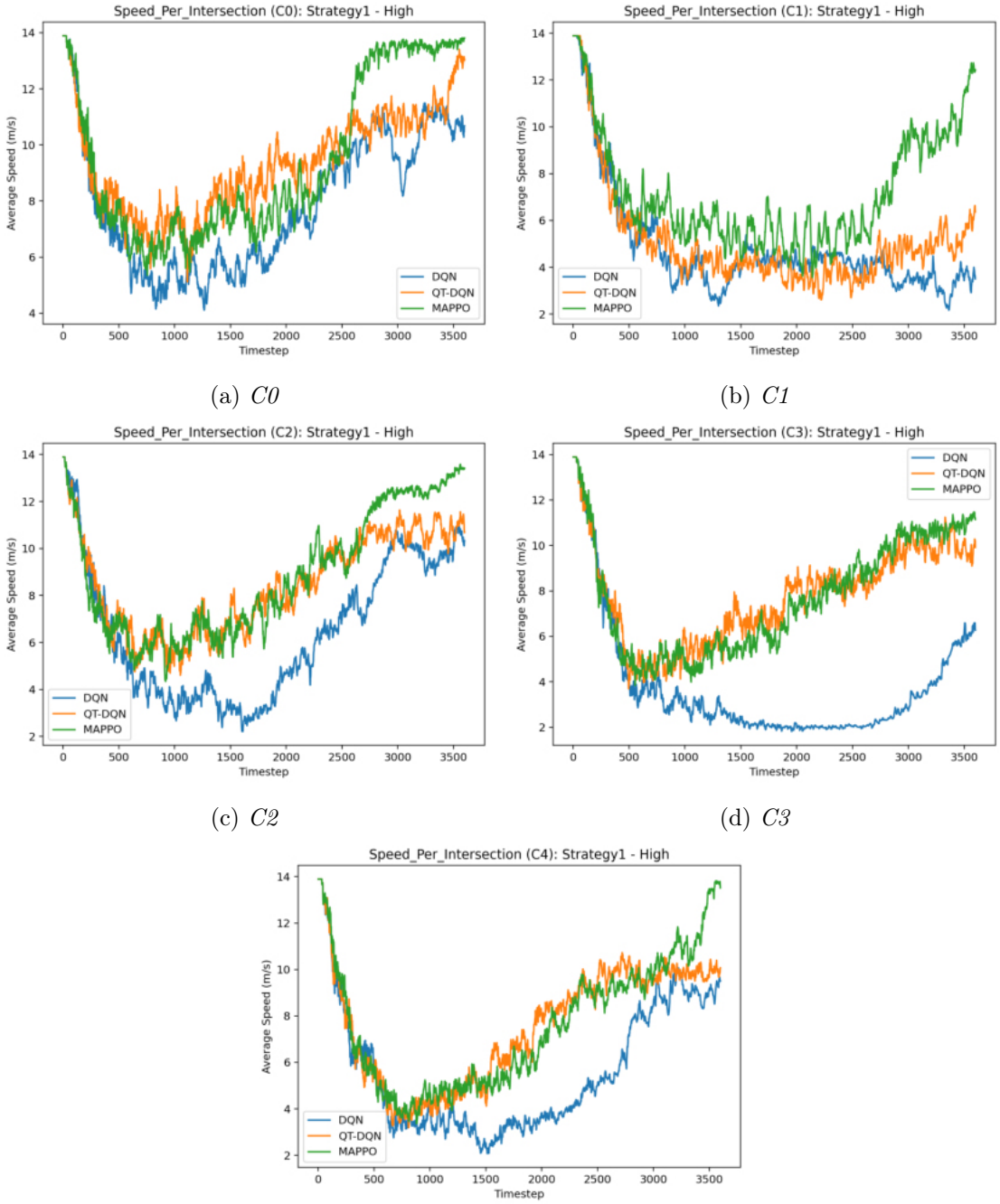
(d) *C3*



(e) *C4*

Figure B.13: *Halting vehicles per intersection (High-demand) — Strategy 1.*

B.2.1.3 Average Speed per Intersection



(a) *C0* (b) *C1*

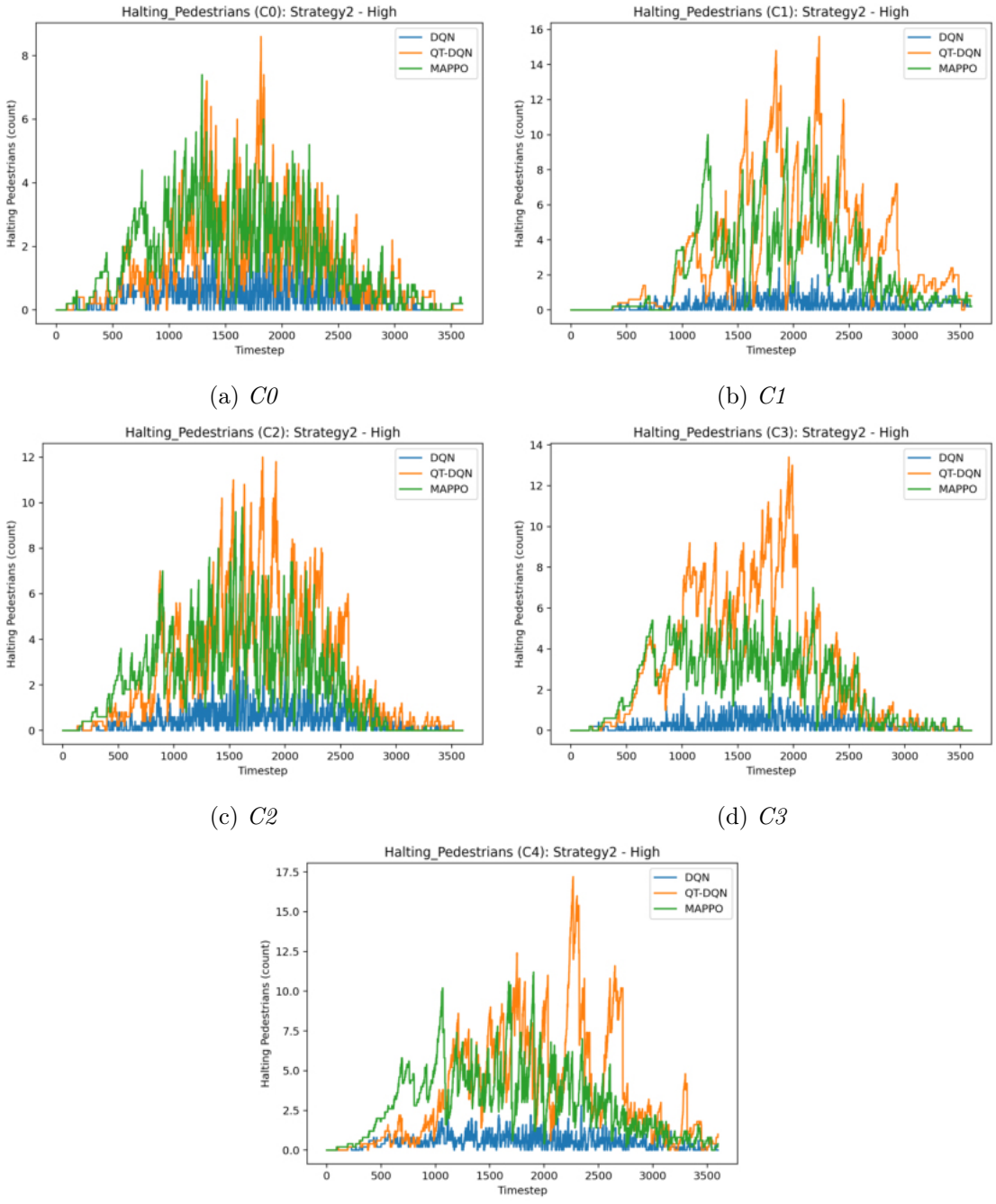
(c) *C2* (d) *C3*

(e) *C4*

Figure B.14: *Average speed per intersection (High-demand) — Strategy 1.*

B.2.2 Strategy 2

B.2.2.1 Pedestrian Halting



(a) *C0*

(b) *C1*

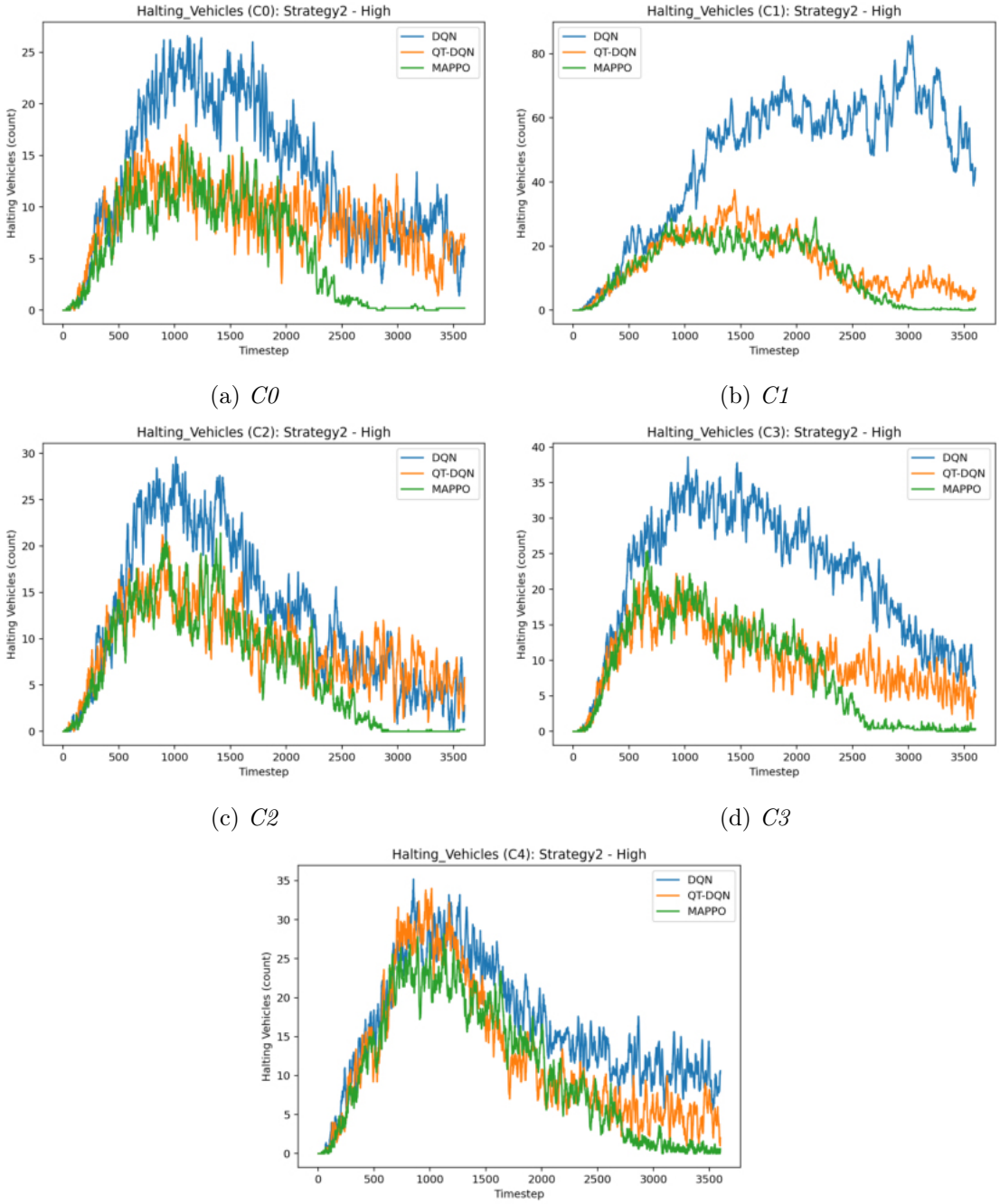
(c) *C2*

(d) *C3*

(e) *C4*

Figure B.15: *Halting pedestrians per intersection (High-demand) — Strategy 2.*

B.2.2.2 Vehicle Halting



(a) *C0*

(b) *C1*

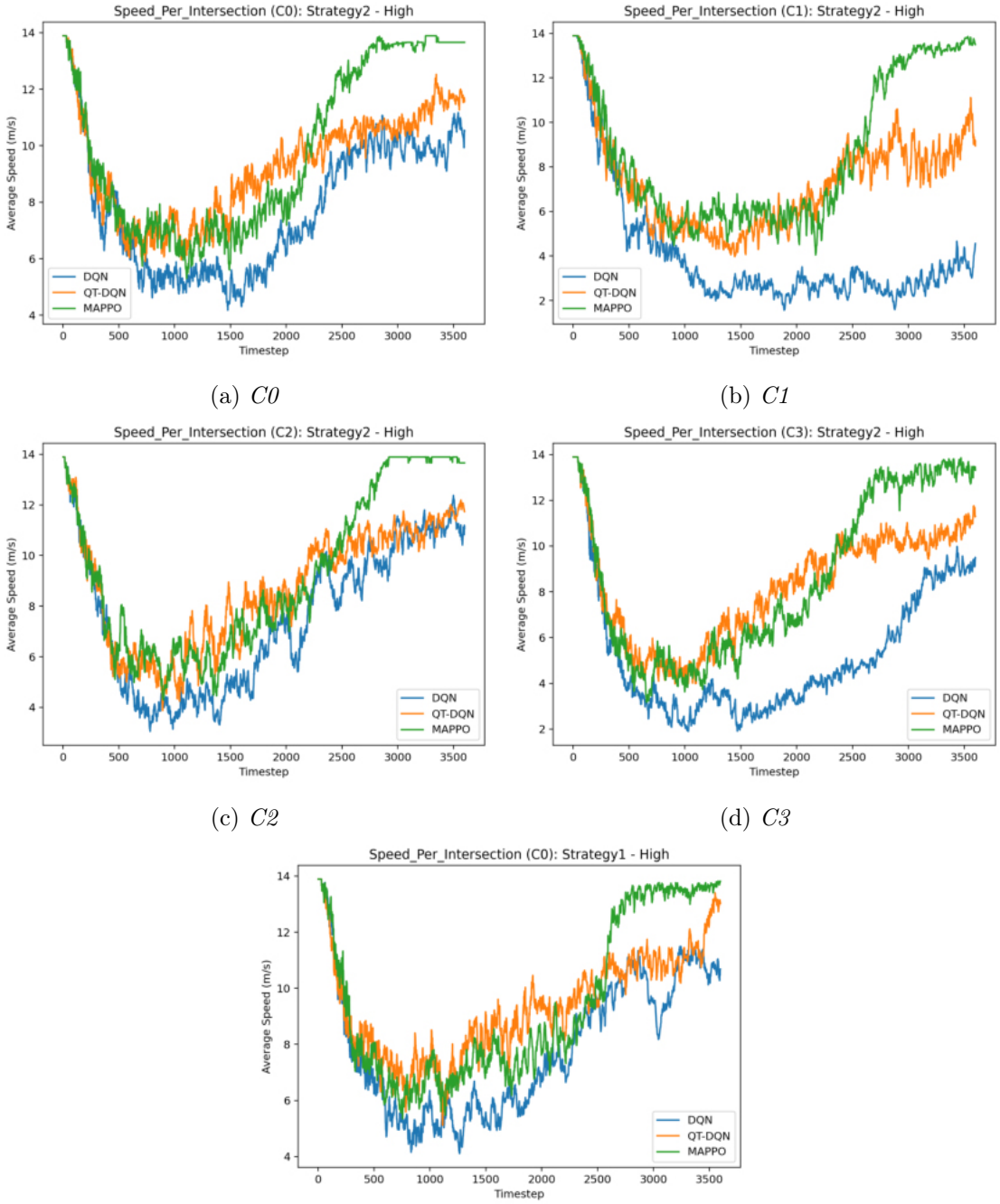
(c) *C2*

(d) *C3*

(e) *C4*

Figure B.16: *Halting vehicles per intersection (High-demand) — Strategy 2.*

B.2.2.3 Average Speed per Intersection



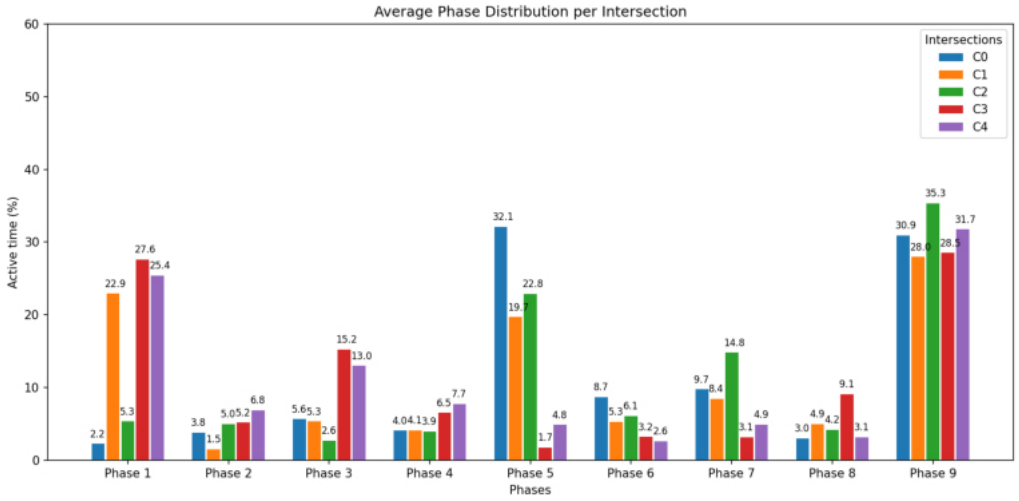
(a) *C0* (b) *C1*

(c) *C2* (d) *C3*

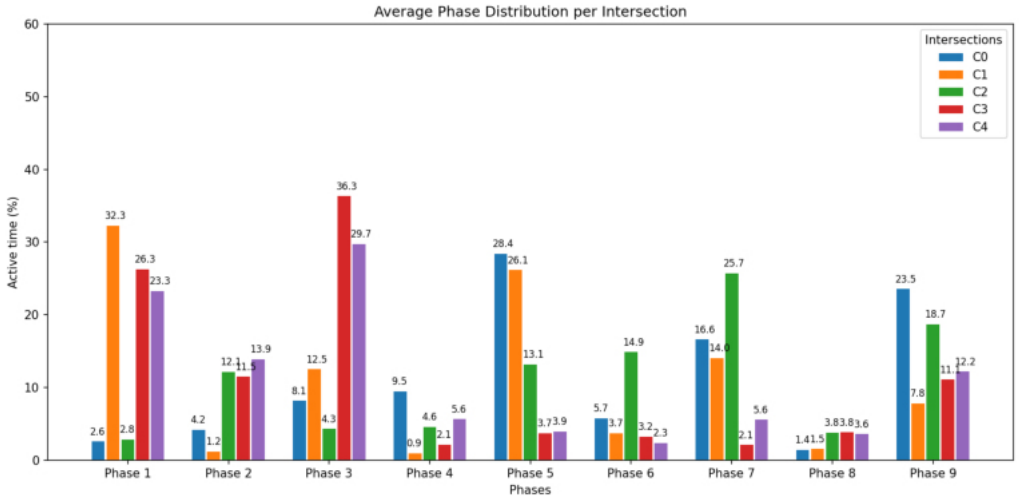
(e) *C4*

Figure B.17: *Average speed per intersection (High-demand) — Strategy 2.*

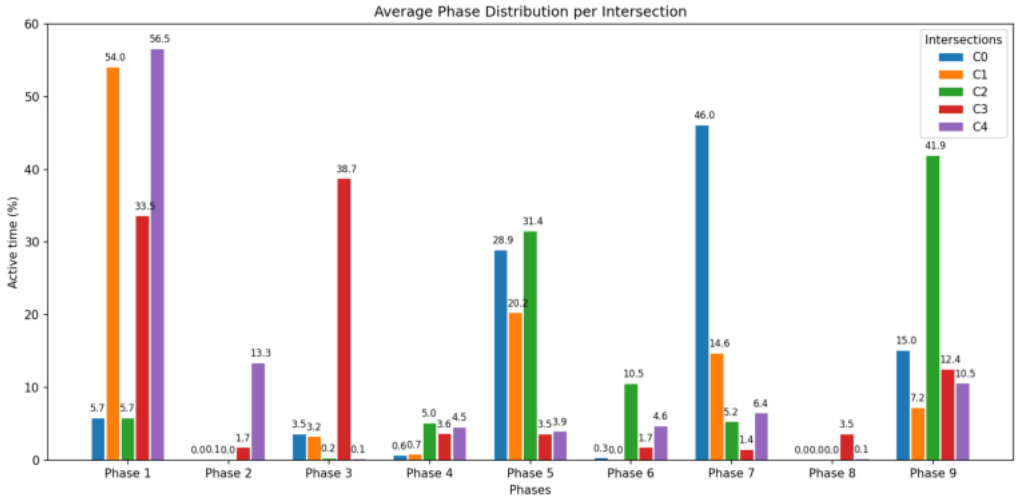
B.2.2.4 Average Phase Distribution



(a) *DQN*



(b) *QT-DQN*

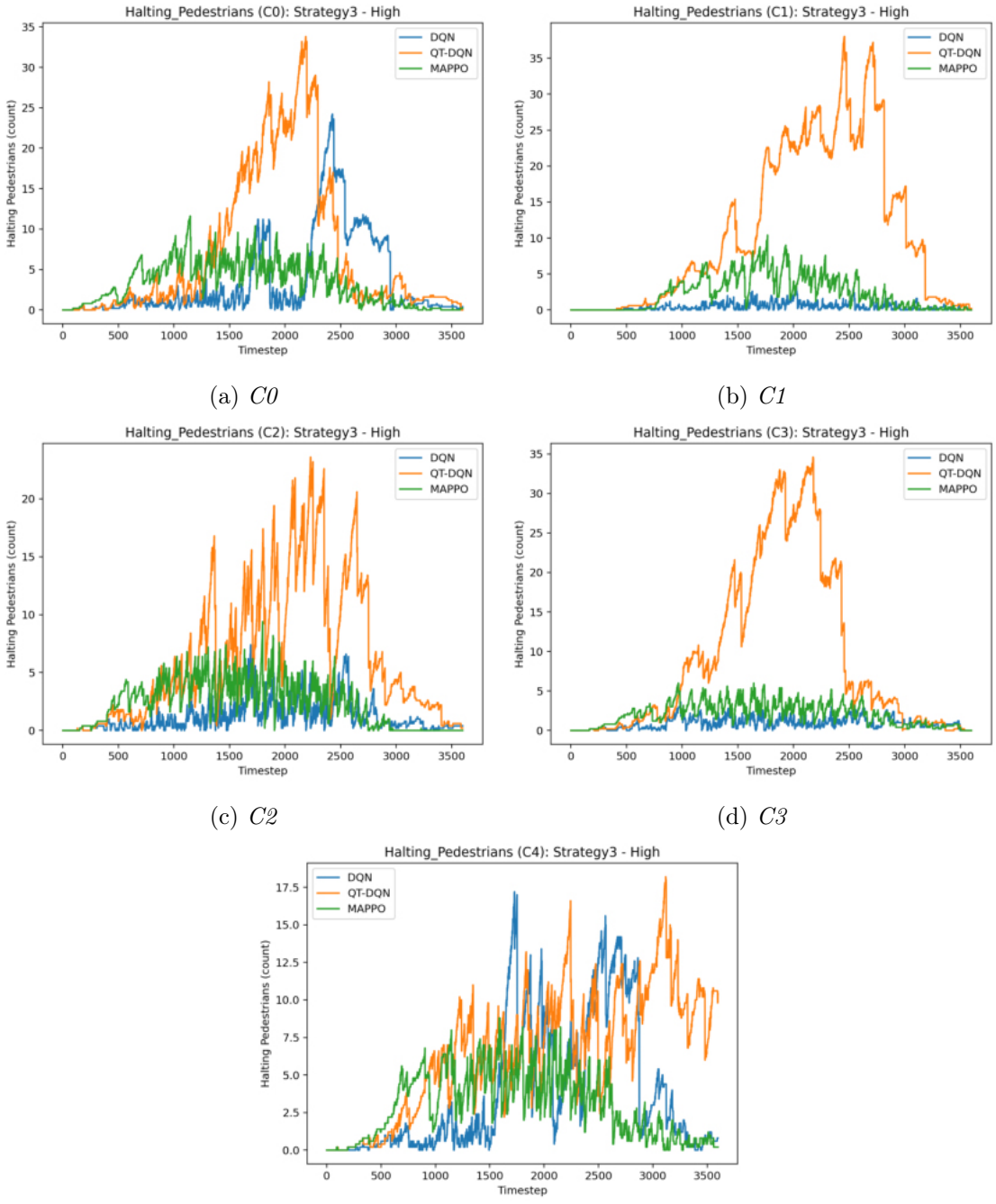


(c) *MAPPO*

Figure B.18: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 2, High scenario).*

B.2.3 Strategy 3

B.2.3.1 Pedestrian Halting



(a) *C0*

(b) *C1*

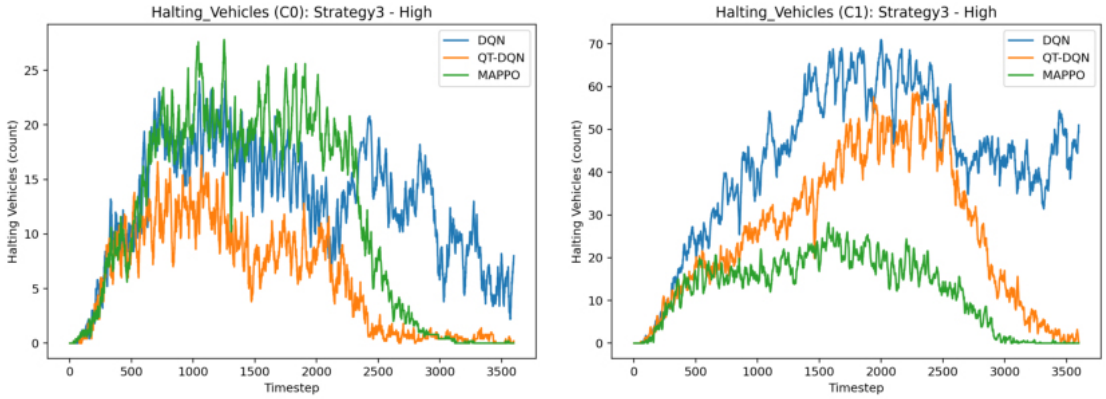
(c) *C2*

(d) *C3*

(e) *C4*

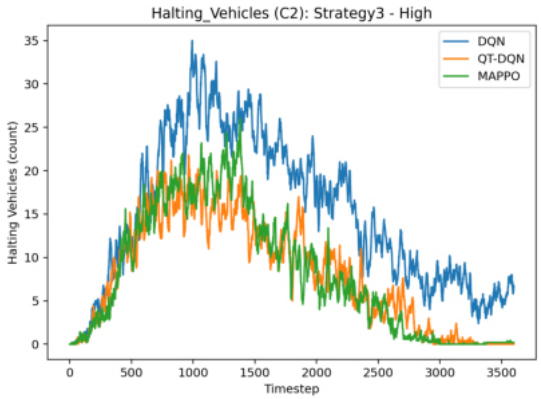
Figure B.19: *Halting pedestrians per intersection (High-demand) — Strategy 3.*

B.2.3.2 Vehicle Halting

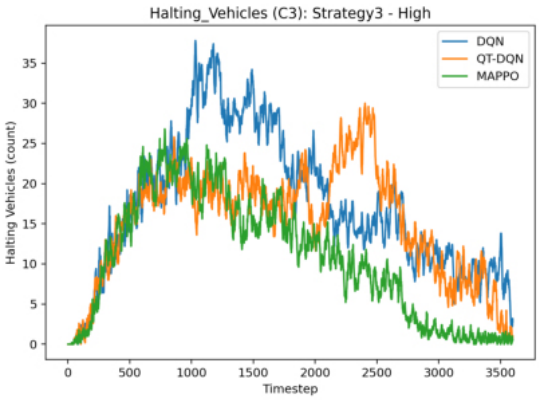


(a) *C0*

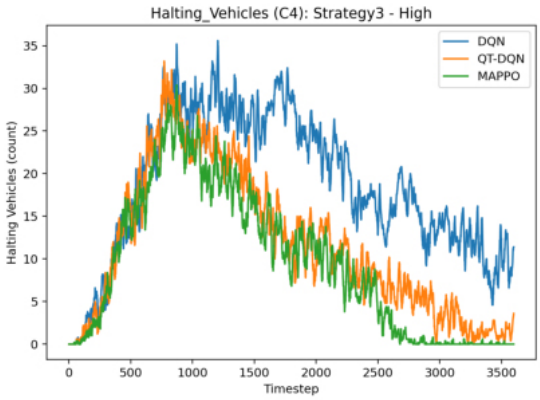
(b) *C1*



(c) *C2*



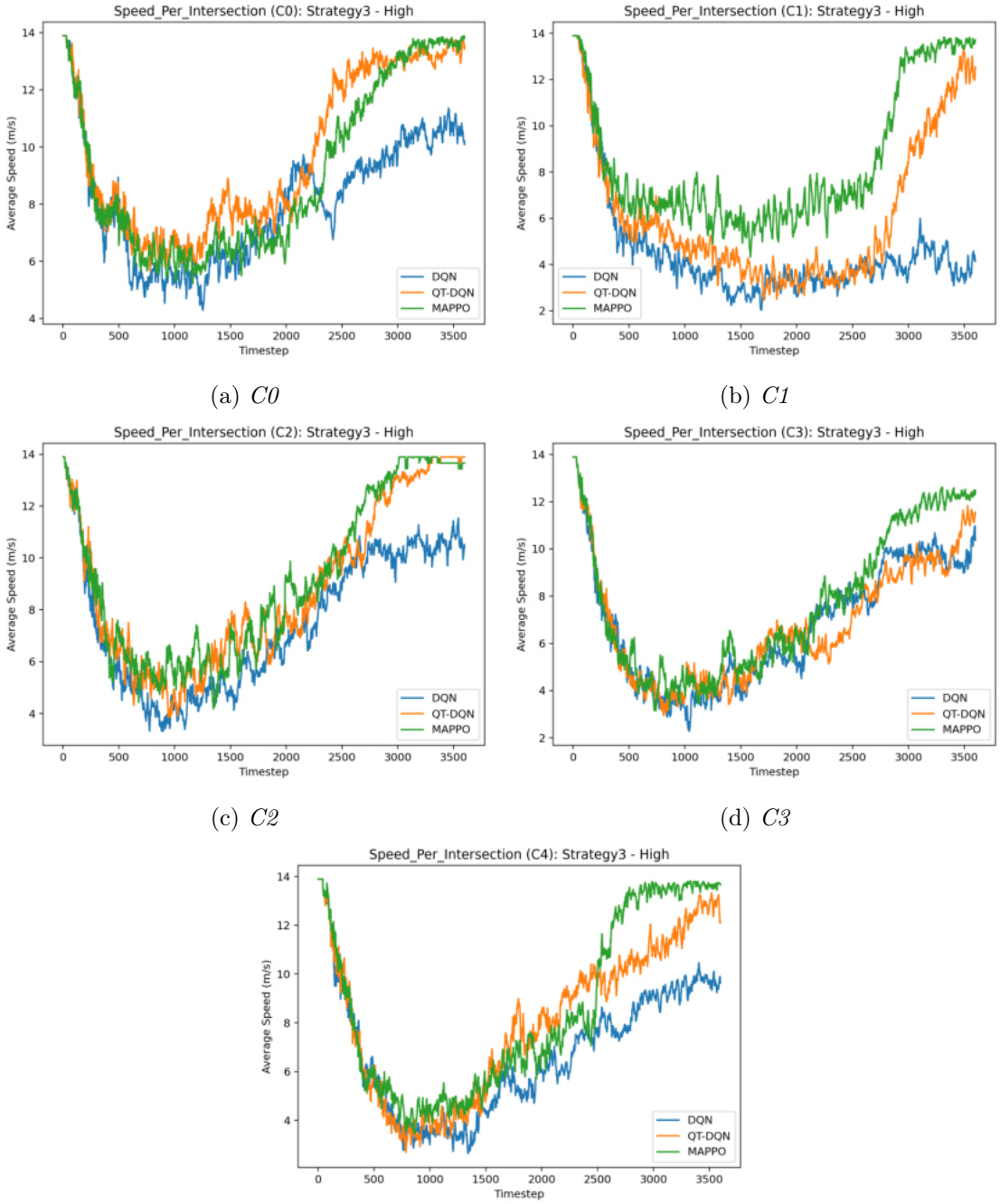
(d) *C3*



(e) *C4*

Figure B.20: *Halting vehicles per intersection (High-demand) — Strategy 3.*

B.2.3.3 Average Speed per Intersection



(a) *C0*

(b) *C1*

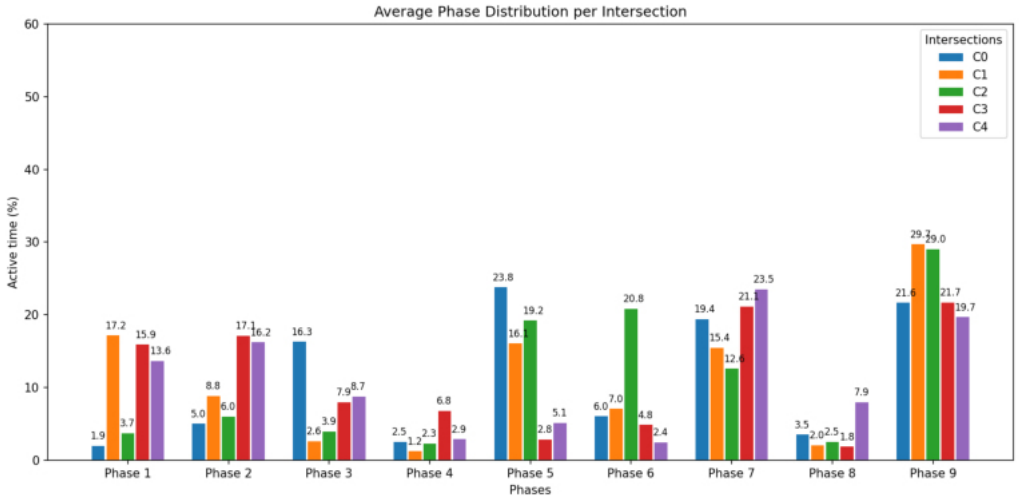
(c) *C2*

(d) *C3*

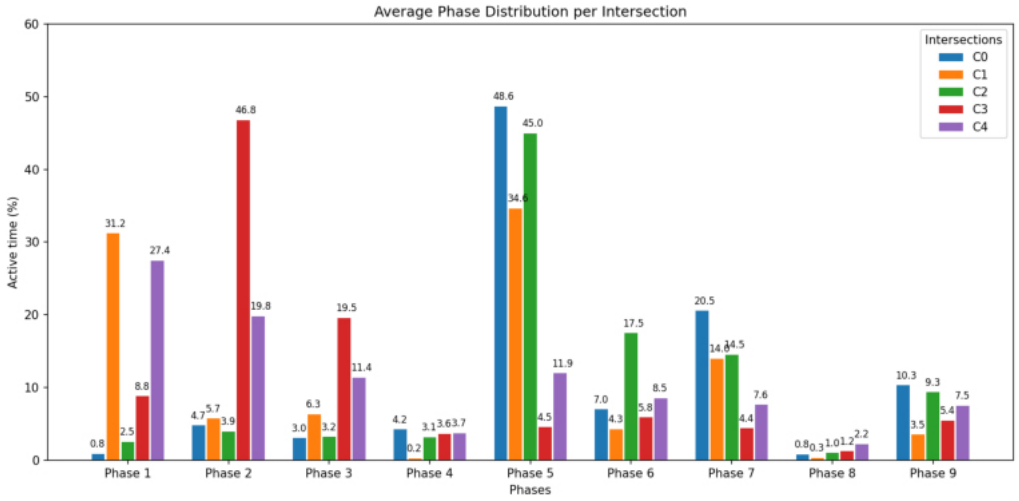
(e) *C4*

Figure B.21: *Average speed per intersection (High-demand) — Strategy 3.*

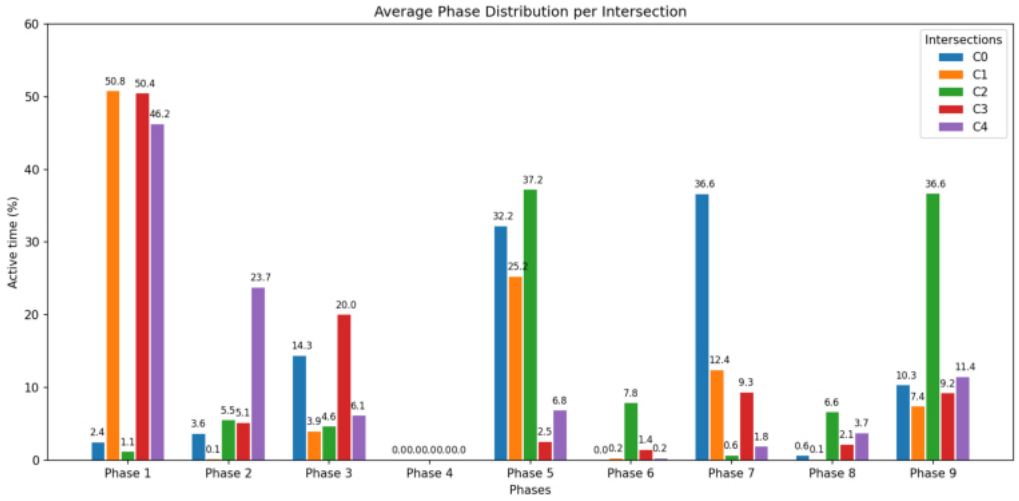
B.2.3.4 Average Phase Distribution



(a) *DQN*



(b) *QT-DQN*



(c) *MAPPO*

Figure B.22: *Average phase distribution for DQN, QT-DQN, and MAPPO (Strategy 3, High scenario).*