



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

School of Computer Science and Statistics

Hierarchical Multi-agent Deep Reinforcement Learning for System-wide Urban Traffic Optimization

Yucheng Shi

Supervisor: Prof. Vinny Cahill

Cosupervisor: Prof. Ivana Dusparic

November 11, 2025

A Thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor in Philosophy

Declaration

I hereby declare that this Thesis is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at <http://www.tcd.ie/calendar>.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at <http://tcd-ie.libguides.com/plagiarism/ready-steady-write>.

I consent / do not consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. **Please consult with your supervisor on this last item before agreeing, and delete if you do not consent**

Signed: Yu cheng Shi

Date: 10/11/2025

Abstract

Rapid urbanization has led to a substantial rise in traffic volumes and the expansion of road networks, creating large and complex urban road transportation systems. Existing traffic-light systems, however, lack the adaptability to effectively manage dynamic traffic conditions. Variations in intersection geometry, lane configurations, and traffic flow can overwhelm these systems, resulting in congestion, lower throughput, unpredictable journey times, and increased emissions and fuel consumption. Consequently, an adaptive and generalized traffic-light-free approach to urban traffic management holds significant potential to improve traffic flow by exploiting the deployment of connected and automated vehicles (CAVs) in large-scale traffic networks.

Effectively managing urban traffic demands a coordinated and adaptive approach, considering the presence of numerous interconnected intersections with distinct geometric layouts and varying traffic volumes. This diversity makes developing a generalized and scalable solution particularly challenging. The large scale of urban road networks requires solutions that ensure not only efficiency at individual intersections but also optimize system-wide performance. Additionally, adapting to real-time traffic variations while maintaining computational efficiency places significant demands on potential solutions.

From a research perspective, urban-scale traffic optimization remains dominated by variants of traffic light control methods, with unsignalized intersection optimization at such scales largely unexplored. Recent studies on unsignalized intersection management are broadly categorized into heuristic-based and learning-based methods, but both face challenges in managing large-scale and heterogeneous networks. Heuristic-based policies struggle with high traffic demand and unforeseen variations, requiring extensive computational resources and being constrained by platoon size limitations. Learning-based approaches leveraging CAVs show promise on isolated intersections but remain insufficiently explored for large-scale scenarios, focusing primarily on basic collision avoidance. This highlights the need for new scalable strategies to manage networks of unsignalized intersections effectively.

This thesis addresses this gap by introducing a hierarchical multi-agent deep reinforcement learning (MADRL) optimization framework for large-scale, unsignalized, and heterogeneous urban road networks. Leveraging the coordination advantages of CAVs, the solution employs a three-tiered control structure to optimize traffic flow. At the lowest level, precise trajectories are calculated for CAVs. The middle layer employs parallel neural Monte-Carlo tree search (PNMCTS) to evaluate intersection crossing schedules simultaneously, selecting the optimal schedule for collision-free crossing. At the highest level, a decentralized MADRL system coordinates multiple intersections across the network, grouping vehicles into platoons of varying lengths to maximize throughput. This integrated approach aims to provide a scalable, efficient and safe solution capable of optimizing traffic flow across heterogeneous urban networks under diverse traffic conditions.

We evaluate the proposed system through a comprehensive set of experiments that assess each level of the hierarchical framework. The evaluation encompasses both homogeneous and heterogeneous intersection networks, along with large-scale simulations of Dublin city's road network. Using zero-shot and few-shot learning, we examine the system's generality and scalability across diverse urban environments.

The experimental results highlight that the proposed hierarchical MADRL framework significantly outperforms traffic-light-based systems, providing notable improvements in average travel times, travel-time reliability, and intersection throughput across diverse urban traffic scenarios. These findings underscore the potential of the hierarchical, multi-agent strategy to reduce congestion, improve intersection performance in large-scale, non-stationary urban environments. The proposed approach advances traffic management by integrating precise trajectory control, PNMCTS cross scheduling, and MADRL for scalable, adaptive, and system-wide optimization, providing a comprehensive solution to the dynamic challenges of future urban traffic systems.

Acknowledgements

Every journey comes to an end, and as I reach the conclusion of this one, I am filled with deep gratitude for all those who have supported me along the way.

First and foremost, I want to thank my parents for their unwavering love, encouragement, and belief in me. Your support has been the foundation of this journey. To my wife—thank you for your love, patience, understanding, and quiet strength. Your presence has carried me through the most challenging moments, and this achievement would not have been possible without you. To our soon-to-be-born child—though you have not yet arrived, your presence has already brought new meaning and hope to everything I do.

To my cat Jinbao—my loyal companion and co-pilot through countless long nights. You are a cool boy, and your calm, steady presence beside me, even in silence, has meant more than words can say.

I am deeply grateful to my supervisor, Professor Vinny Cahill, for his mentorship and for giving me the opportunity to pursue my doctoral research in the Distributed Systems Group (DSG) at Trinity College Dublin. I also sincerely thank my co-supervisor, Dr. Ivana Dusparic, for her guidance, insight, and constant encouragement throughout the course of my PhD. It has been a privilege to work under your supervision and to grow through your mentorship.

I gratefully acknowledge the Centre for Research Training in Artificial Intelligence (CRT-AI) and Science Foundation Ireland (SFI) for their funding and support, without which this research would not have been possible.

Finally, I am thankful to the entire DSG research group and the academic community at Trinity College Dublin. The collaborative atmosphere, intellectual freedom, and shared curiosity have enriched this journey in ways beyond words.

Years from now, I may not remember every detail of this thesis, but I will remember the people, the stories, and the moments that shaped this journey.

Contents

1	Introduction	1
1.1	Research Motivations	2
1.1.1	The Challenge of RL-based Traffic control Methods	2
1.1.2	The Challenge of System-wide Traffic Optimization	3
1.1.3	The Challenge of Heterogeneity in Large-scale Unsignalized Traffic Networks	4
1.1.4	Conclusion	5
1.2	Research Objective and Questions	5
1.3	Thesis Approach and Contributions	6
1.4	Assumptions	7
1.5	Thesis Roadmap	9
2	Background	11
2.1	Connected and Automated Vehicles	12
2.2	Markov Decision Processes and Reinforcement Learning	13
2.2.1	Value-Based Methods and Deep Q-Networks (DQN)	14
2.2.2	Policy-Based Methods and the Actor-Critic Framework	15
2.2.3	Proximal Policy Optimization (PPO)	15
2.3	Multi-Agent Deep Reinforcement Learning	17
2.3.1	Key Concepts in MARL	17
2.3.2	Challenges in MARL	17
2.3.3	Training Frameworks for Multi-Agent Reinforcement Learning	18
2.4	Hierarchical Reinforcement Learning	19
2.4.1	Structure of HRL	19
2.4.2	Advantages of HRL	19
2.5	Monte-Carlo Tree Search and Neural Monte-Carlo Tree Search	20
2.5.1	Monte-Carlo Tree Search	21
2.5.2	Neural Monte-Carlo Tree Search	22
2.5.3	Applications of MCTS and NMCTS	24
2.6	Traffic Simulation and Control Frameworks	25
2.6.1	Simulation of Urban Mobility	25
2.6.2	Traffic Control Interface	26
2.7	Summary	26

3	State-of-the-art Review	28
3.1	Evolution of Signalized Traffic Control Systems	28
3.1.1	Deployed Traffic Control Systems	28
3.1.2	Advanced Traffic Control Systems	29
3.1.3	Summary	30
3.2	Taxonomy of Unsignalized Traffic Control Systems	30
3.2.1	Modelling Unsignalized Intersections	30
3.2.2	Architectures for Managing Vehicle Crossing at Unsignalized Intersections . .	32
3.2.3	Optimizing Crossing on Individual Intersections	39
3.2.4	Optimizing Crossing on Multiple Intersections	43
3.3	Research Gaps	50
3.3.1	Gap 1: Lack of Hierarchical Frameworks for Decentralized Traffic Management	50
3.3.2	Gap 2: Absence of Standardized Intersection Representation Models	50
3.3.3	Gap 3: Insufficient Integration of Individual and System-Wide Objectives . . .	50
3.3.4	Gap 4: Limited Training Methods for Large-Scale Multi-Agent Systems	50
3.3.5	Scheduling Under Specific Intersection Geometries and Traffic Flows	51
3.4	Summary	51
4	Overall Design of the Hierarchical Multi-Agent Deep Reinforcement Learning Frame- work	52
4.1	Motivation for and Structure of the Proposed HRL Framework	52
4.1.1	Motivation	53
4.1.2	Proposed HRL Framework	54
4.2	Transforming Heterogeneous Intersections into a Uniform Representation	55
4.2.1	Automated Transformation of Heterogeneous Intersection Geometries	55
4.2.2	Representing Continuous Traffic Flow	56
4.3	Balancing Short-Term and Long-Term Objectives Through Hierarchical Rewards . . .	56
4.3.1	Schedule Continuous Vehicle Crossings at Unsignalized Intersections	57
4.3.2	Hierarchical Rewards	57
4.4	System-wide Optimization	58
4.4.1	Platoon Method	58
4.4.2	Multi-Agent Proximal Policy Optimization	59
4.5	Low-Level Vehicle Trajectory Control with SUMO	59
4.6	Summary	59
5	Optimizing Intersection Scheduling	61
5.1	Intersection Transformation Model	62
5.1.1	Example Transformation of Intersection Dynamics	62
5.1.2	Discrete-Time State Space Model	63
5.2	Learning Environment Design	67
5.2.1	Observation Space	67
5.2.2	Action Space	68
5.2.3	Reward Model	69

5.3	Training Method	70
5.3.1	Parallel Neural Monte-Carlo Tree Search	70
5.4	Summary	76
6	Network-wide Traffic Optimization	78
6.1	Training Scheduling Policy for Large Heterogeneous Networks	79
6.1.1	Automatic Detection of Collision Areas in Unsignalised Intersections	79
6.1.2	Geometry-Agnostic Learning for Scalable Policy Generalization	80
6.2	Learning Environment Design for Network-wide Traffic Optimization	83
6.2.1	Traffic Pressure and State Representation	83
6.2.2	Lane-Level Traffic Pressure Representation	85
6.2.3	Intersection State Representation	86
6.2.4	Action Space	87
6.2.5	Reward Model	87
6.3	Training Method	89
6.3.1	Neural Network Structure and Optimization Strategy	90
6.3.2	Training Schema	91
6.4	Summary	92
7	Implementation of Vehicle Trajectory Control in SUMO	93
7.1	CAV Parametrization for Unsignalized Urban Intersections	93
7.2	Vehicle Speed Control Problem for Unsignalized Intersection Crossing	94
7.2.1	Notation	96
7.3	Multi-segment Acceleration Control Algorithm	97
7.4	Determining the Duration of First-Segment Acceleration	99
7.5	Summary	101
8	Evaluation	102
8.1	Evaluation Baselines and Metrics	104
8.1.1	Baselines for Single Unsignalised Intersection	104
8.1.2	Baselines for Homogeneous Multi-Intersections	105
8.1.3	Baselines for Heterogeneous Urban Networks	105
8.1.4	Metrics	106
8.2	Evaluation of the Intersection Scheduling Method	107
8.2.1	Evaluation of PNMCTS Performance on Single intersection	107
8.2.2	Evaluation of PNMCTS Performance on 3×3 Grid Network	108
8.2.3	Ablation Experiments	109
8.2.4	Summary of Intersection Scheduling	124
8.3	Evaluation of Traffic Network Optimization	124
8.3.1	Evaluation of HRL Performance on Single intersection	125
8.3.2	Evaluation of HRL-MAPPO Performance on 3×3 Grid Network	128
8.3.3	Evaluation of HRL-MAPPO Performance on Dublin City Network	128
8.3.4	Ablation Study of HRL-MAPPO	136

8.4	Analysis and Insights	139
8.5	Summary	142
9	Conclusion	144
9.1	Summary of Technical Approach	144
9.1.1	Hierarchical Reinforcement Learning Framework for Urban Traffic Management	144
9.1.2	Decentralized Multi-Agent Reinforcement Learning with MAPPO for Traffic Coordination	145
9.1.3	Standardized Intersection Representation with Automated Collision Detection	145
9.1.4	Platoon Scheduling with PNMCTS	146
9.1.5	Zero-Shot and Few-Shot Fine-Tuning	146
9.2	Answering the Research Questions	146
9.3	Limitations and Future Work	148

List of Figures

4.1	The system optimization agent and intersection scheduling agent are deployed in a layered hierarchy at intersections in the urban network.	54
4.2	An abstract representation of each layer in the proposed architecture.	54
4.3	A detailed architecture design for the proposed decentralized hierarchical multi-agent deep reinforcement learning framework.	60
5.1	An intersection with four approaching platoons is illustrated on the left. The representative board corresponding to this scenario is given on the right. Entry and exit times for collision areas are shown on the time line (X-axis), while the Y-axis shows platoon numbers. As platoon 1 is moving forward, it will cross collision areas C→B→A in that order, with the occupancy time of the corresponding collision area being mapped to time blocks (shown stacked for each platoon for readability). The occupancy time of collision areas C, B, and A by platoon 1 overlap because the platoon is longer than the gap between the collision areas. We can observe from the illustration that, for example, without a scheduling intervention, platoons 1 and 4 will collide in collision area A, Platoons 3 and 4 will collide in collision area D, as they would occupy it simultaneously.	63
5.2	A potential solution to a conflicting board.	64
5.3	Transformation board represents a busy intersection that is executing a previous schedule (the upper board) and cannot be rescheduled. The new approaching platoons have potential to collide with the previous schedule at collision areas A and B.	64
5.4	A potential solution to the conflicting board. The new schedule (for platoon 5 and 6) must ensure a compatible solution to the solved schedule (for platoon 1,2,3 and 4).	65
5.5	DTSSM design for intersection crossing.	66
5.6	Grid representation of transformation board according to Figure 5.1.	67
5.7	Transition of a board from its initial state to the solved state by applied a discrete action with 3 seconds delay.	68
5.8	A demonstration of MCTS rollout from an initial board.	69
5.9	Dual-branch neural network architecture for board processing and action selection.	70
5.10	Training with parallel neural Monte-Carlo tree search.	72
5.11	Distribution of steps to resolve busy boards.	75
5.12	Short-path MCTS rollout: branch a for empty boards, branch b for busy boards.	76

6.1	An urban geometry-agnostic learning approach. Composite-policy training process (left), evaluation with fine-tuned policy (top-right), and zero-shot evaluation (bottom-right).	81
6.2	The definition of intersection traffic pressure. The green circle at each intersection serves as a trigger for initiating intersection scheduling, when the incoming platoons are determined incompatible.	84
6.3	A real intersection in Dublin city contains multiple virtual lanes. This example shows how sub-lane-based traffic pressure works at a deformed intersection.	85
6.4	Centralised Training with Decentralised Execution (CTDE) framework.	89
6.5	High-level neural network structure.	90
7.1	The multi-segment trajectory control process.	95
7.2	Two-segment formulation of the slot synchronization process.	97
8.1	Evaluation Roadmap of the Proposed HRL-MAPPO Traffic Network Optimization Framework.	103
8.2	A homogeneous 3×3 grid intersection network.	105
8.3	Dublin city centre traffic network.	106
8.4	Crossing time comparison between FIFO and PNMCTS method at, (a) Low-traffic intersections and (b) High-traffic intersections. All solutions provided by PNMCTS outperforms the FIFO in both scenarios.	108
8.5	Performance comparison between different NN structures.	111
8.6	Ablation study comparison on schedule quality Evolution. Median rewards and corresponding variances are plotted for different enhancements across training steps.	112
8.7	Median rewards with standard deviations using standard NMCTS	113
8.8	Median rewards with standard deviations using PNMCTS-RE	113
8.9	Median rewards with standard deviations using PNMCTS-REC	114
8.10	Median rewards with standard deviations using PNMCTS-RECS	114
8.11	Total number of solved board in the evaluation boards.	115
8.12	Times each board is solved repeatedly using standard NMCTS	116
8.13	Times each board is solved repeatedly using PNMCTS-RE	116
8.14	Times each board is solved repeatedly using PNMCTS-REC	117
8.15	Times each board is solved repeatedly using PNMCTS-RECS	117
8.16	Comparison between optimal and sub-optimal schedules. (a) An optimal solution use 0.9 second.(b) A sub-optimal solution use 0.2 second. Note that the Last action is explicitly different as the PNMCTS policy performs a finer-grained action leading to a better solution.	118
8.17	An optimal crossing schedule provided by a PNMCTS agent at a busy intersection from top to bottom, platoon delay sequence: initial board $\rightarrow$ platoon1 $\rightarrow$ platoon0 $\rightarrow$ platoon2 $\rightarrow$ platoon0. The total time usage is 1.2 seconds which is longest among all evaluation boards.	119
8.18	Traffic flows are evenly inserted from the marginal road to simulate a correct traffic patterns.	120

8.19	Progressive replacement of fixed-time traffic light controllers with agent-based controllers across the network. Controlled intersections reduce queue build-up in incoming lanes, as evidenced by reduced travel times (highlighted in green). A fully agent-controlled network can achieve near free-flow travel conditions.	121
8.20	Single agent deployment leads to more downstream congestion.	122
8.21	(a): Average travel time and (b): total throughput of the network for scenarios 1-6.	122
8.22	Progressive replacement of fixed-time traffic light controllers with agent-based controllers across the network. Schedule reward and inference time details.	123
8.23	Cyclic square functions for traffic flow in various directions.	126
8.24	Sine wave functions for traffic flow in various directions.	127
8.25	Comparison of total throughput, average travel time, and travel time reliability. Higher extra travel time indicates lower travel reliability.	129
8.26	Comparison of detected collision areas and the real-world street view of junction 'road-108'.	131
8.27	Automatically detected collision areas in Dublin city centre using DBSCAN clustering algorithm.	132
8.28	Traffic-light control on Dublin city network with 30% CAV penetration.	133
8.29	Proposed MAPPO zero-shot evaluation on Dublin city network.	133
8.30	Fine-tune area in Dublin city network.	134
8.31	Proposed fine-tune evaluation on Dublin city network.	134
8.32	Overall episode rewards for different values of α . MAPPO refers to collaborative policies incorporating neighbour rewards at varying levels, controlled by parameter α , while ISO-PPO utilizes only individual observations and local rewards.	136
8.33	Per-agent episode rewards for different values of α . MAPPO integrates cooperative observation with varying levels of neighbour rewards, while ISO-PPO relies solely on local observations and rewards.	137
8.34	Comparison of ATT and TTP across different platoon lengths and control strategies.	138

List of Tables

2.1	Urban model results for different CAV penetration rates [U.K. Government Service, 2016].	12
3.1	Summary of CAV Conflict Modelling Methods	32
3.2	Summary of Centralized Intersection Management Frameworks	35
3.3	Summary of Decentralized Intersection Management Frameworks	38
3.4	Summary of HRL Frameworks in Traffic Management	40
3.5	Summary of Heuristic-Based Optimization Methods for Isolated Intersection Management	42
3.6	Summary of RL-Based Approaches for Intersection Management	44
3.7	Summary of Heuristic-Based Optimization on Multi-Intersection Scenarios	45
3.8	Summary of Unsignalized Intersection Management Frameworks (Part 1)	47
3.9	Summary of Unsignalized Intersection Management Frameworks (Part 2)	48
3.10	Summary of Unsignalized Intersection Management Frameworks (Part 3)	49
8.1	Travel time and throughput comparison in a 3×3 grid network with 3 runs	109
8.2	Success rate comparison at synthetic boards	118
8.3	Low, moderate, and high traffic flow rates (in vehicles per hour) for a single unsignalized intersection.	125
8.4	ATT Comparison of State-of-the-Art Baselines for Single Unsignalized Methods. Lower the better, and the best results are in bold.	125
8.5	Key simulation parameters for the Dublin city environment.	130
8.6	Comparison of Different Methods on Traffic Management Metrics. Abbreviations: TTR – Travel Reliability, TT – Total Throughput, TTD – Total Travel Distance, ATT – Average Travel Time, WT – Waiting Time, WC – Waiting Count.	135

1 Introduction

Optimizing large-scale urban traffic networks has been a critical area of transportation research for decades [Keyvan-Ekbatani et al., 2015, Rasheed et al., 2020]. Early approaches, such as fixed-time traffic light control [Serafini and Ukovich, 1989], and more recent adaptive systems [Shinde, 2017], have formed the backbone of urban traffic management. Despite these advancements, traditional traffic-light systems often fail to address the complexities of modern urban environments, resulting in inefficiencies and congestion. According to the Federal Highway Administration, signalized intersections significantly contribute to delays and fuel consumption, especially during peak hours [Federal Highway Administration, 2004]. Frequent idling and deceleration at these intersections further exacerbate emissions and other inefficiencies [Hughes et al., 2010]. For example, the Texas Transportation Institute estimates that in 2022, congestion in the USA resulted in 8.5 billion hours of total delay, 3.3 billion gallons of wasted fuel, and a total economic cost of \$224 billion [Texas Transportation Institute, 2023]. These figures highlight the urgent need for intelligent and adaptive traffic management solutions to improve travel-time reliability and efficiency.

The emergence of connected and automated vehicle (CAV) technologies presents a transformative opportunity for urban traffic management. By enabling dynamic, real-time coordination between vehicles and infrastructure, CAV technologies introduce a new paradigm for intelligent transportation systems [Lee and Hess, 2020, Keskin et al., 2020, de Dios Ortúzar, 2021].

Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, combined with autonomous driving capabilities, promise significant improvements in traffic flow and efficiency. These advancements hold the potential to reduce delays, minimize fuel consumption, and improve the overall travel experience [Zhao et al., 2023, Wang et al., 2018].

Leveraging the coordination capability of CAVs, current research on unsignalized networks can be broadly categorized into heuristic-based methods and reinforcement learning (RL)-based methods. Heuristic-based methods have been applied to networks of up to fifty homogeneous intersections [Hausknecht et al., 2011, Vasirani and Ossowski, 2011, Tlig et al., 2014, Pei et al., 2021, Iwase et al., 2022, Zhang and Zhou, 2018]. While these methods demonstrate scalability, most of them still face challenges such as high computational costs, and limited generalizability across diverse intersection layouts and varying traffic volumes. RL-based methods have mainly targeted individual intersections or small-scale systems, focusing on optimizing intersection crossing. For example, early studies integrated vehicle-specific dynamic and kinematic constraints into deep RL models [Demir and Sezer, 2018] on a single intersection. Subsequent advances combined MCTS with DDQN for improved action value approximations, enhancing safety

and efficiency [Weingertner et al., 2020]. More recently, RL-integrated sequential optimization has further refined intersection navigation, promoting smoother and safer manoeuvres [Tallapragada et al., 2023].

To address these challenges, this thesis explores novel methodologies leveraging CAV technologies to optimize intersection management in large-scale, heterogeneous unsignalized urban traffic networks. Specifically, it focuses on hierarchical reinforcement learning (HRL) and multi-agent reinforcement learning (MARL) to develop scalable and adaptive frameworks for intelligent urban mobility. These approaches are integrated with the aim of overcoming the limitations of existing heuristic and RL-based methods by enhancing coordination, generalization, and scalability.

This chapter presents the motivation, objectives, and overarching approach for this thesis. It begins by discussing the limitations of traditional traffic management methods and the opportunities presented by RL, MARL, and HRL. The research objectives and questions driving the development of the proposed framework are then introduced, followed by a summary of the key contributions of this work. Finally, the chapter provides an overview of the thesis roadmap.

1.1 Research Motivations

While RL and MARL offer potential solutions for real-time and scalable traffic optimization, their application to urban traffic management still faces numerous open questions and challenges. HRL, with its ability to decompose tasks into manageable sub-problems, provides an additional layer of flexibility for addressing the intricacies of traffic networks. This section outlines the motivation behind exploring advanced RL-based methods, highlights the limitations of existing heuristic and RL-based approaches, and identifies the critical research gaps in leveraging RL, MARL, and HRL for managing heterogeneous unsignalized intersections and large-scale traffic networks. By addressing these gaps, the thesis seeks to contribute to the development of efficient, scalable, and adaptive frameworks for intelligent traffic management systems.

1.1.1 The Challenge of RL-based Traffic control Methods

Effective system-wide optimization of large-scale urban traffic networks requires advanced strategies capable of addressing the high variability and unpredictability inherent in these systems. RL is a branch of machine learning where an agent learns optimal policies by interacting with an environment through trial and error [Sutton, 2018]. Unlike supervised learning, RL does not rely on labelled datasets; instead, it uses a reward signal to guide the agent's learning process. This makes RL particularly suitable for dynamic and complex environments, such as large-scale traffic networks, where predefined rules or labelled data are often unavailable.

In large-scale traffic network management, RL offers several advantages. By allowing agents to learn adaptive control strategies based on real-time traffic conditions, RL has been demonstrated to reduce congestion, and improve overall traffic flow efficiency [Yau et al., 2017]. Moreover, RL's capacity to handle high-dimensional state spaces makes it well-suited for managing complex systems where numerous variables, such as vehicle density and travel time, must be considered simultaneously. These capabilities highlight RL's potential to transform traditional traffic

management approaches. However, single-agent RL and centralized RL approaches often struggle with scalability as the number of intersections or agents increases, resulting in slower convergence and suboptimal policies [Yau et al., 2017, Wu et al., 2021a]. The large state and action spaces inherent in urban traffic systems further complicate optimization and impose a significant computational burden.

To address these limitations, MARL has emerged as a promising framework. MARL extends RL to scenarios involving multiple agents, enabling decentralized control and coordination across a traffic network [Sarker, 2023, Spatharis and Blekas, 2024]. Each agent represents an intersection controller, learning policies independently while interacting with other agents to achieve system-wide optimization. This decentralized nature allows MARL to scale more effectively, reducing computational complexity compared to centralized approaches. Furthermore, MARL facilitates coordination among agents, enabling adaptive and cooperative traffic management strategies.

Despite its advantages, MARL introduces its own set of challenges. Coordination among multiple agents can be difficult as each agent may have conflicting objectives, hindering global optimization [Noaen et al., 2022]. One of the primary challenges in MARL is the **moving target** problem. This issue arises because agents learn concurrently within a shared environment, creating a dynamic setting where each agent's policy is influenced not only by its own interactions but also by the constantly evolving strategies of other agents. As a result, the environment becomes non-stationary, making it difficult for agents to converge on stable and effective policies.

In traffic management scenarios, this problem is particularly evident. For instance, when an agent, such as a vehicle or intersection controller, adjusts its actions to minimise travel time, other agents respond by altering their own paths or scheduling policies. These changes can render the initial agent's optimised policy ineffective, causing a conflicting effect that complicates the overall optimization process.

Recent MARL research has proposed several strategies to address the moving target problem:

- **Centralised Training with Decentralised Execution (CTDE):** Agents are trained using collective information in a centralised manner but operate independently during execution. This approach helps agents leverage shared knowledge while maintaining autonomy [Yu et al., 2022, Iqbal et al., 2021].
- **Stabilisation Techniques:** Methods like experience replay with prioritised sampling help mitigate non-stationarity and enhance learning stability [Papoudakis et al., 2021].

In summary, while single RL provides a robust foundation for optimizing unsignalized traffic flows, its limitations in managing large-scale systems necessitate the use of MARL. Nevertheless, MARL still presents several challenges that require careful design and significant effort to develop smart, flexible, and effective urban traffic management systems.

1.1.2 The Challenge of System-wide Traffic Optimization

Optimizing urban traffic networks requires a layered approach with decisions ranging from controlling individual vehicles to managing network-wide flow. HRL, which breaks complex tasks

into smaller sub-tasks across multiple levels, combined with MARL, offers a promising solution [Sutton et al., 1999b]. In HRL, agents learn policies at different scales—high-level strategies and low-level actions—making it suitable for large-scale cyber-physical systems like traffic networks [Bazzan, 2009]. This combination allows agents at intersections to work together while addressing specific goals, such as reducing delays or improving throughput.

Designing an effective hierarchical structure for HRL and MARL in traffic management depends on smart task breakdown. If control is too fine, i.e., managing every vehicle’s path, it becomes too costly to compute [Dayan and Hinton, 1993]. But if tasks are too broad, key details at intersections—like vehicle interactions—might be missed, hurting overall performance [Shikunov and Panov, 2020]. Finding the right balance is still an open challenge, as noted in studies on hierarchical traffic control [Yildirimoglu et al., 2018].

Another issue is balancing goals across levels. Creating reward functions that handle both short-term wins (e.g., clearing an intersection) and long-term aims (e.g., network flow) is complex. Research on hierarchical traffic signal control shows this tension, with no standard solution yet for large networks [Yildirimoglu et al., 2018]. Studies in related fields, like robotics, suggest HRL can align goals using shared rewards, but applying this to traffic remains underexplored [Nachum et al., 2018].

Urban traffic also changes constantly, so HRL-MARL systems must adapt quickly. High-level plans and low-level controls need to be synchronized to handle shifting patterns, like rush hour spikes. This adaptability is key, and early work on HRL in traffic shows it can respond to changes, though scaling it to large-scale networks is still challenging [Bazzan, 2009].

1.1.3 The Challenge of Heterogeneity in Large-scale Unsignalized Traffic Networks

Urban traffic networks differ widely in size, layout, and flow, making optimization difficult. Unlike signalized intersections with fixed light timings, unsignalized ones rely on vehicle interactions, which are harder to predict [Board, 2010]. In large networks with multiple lanes, mixed vehicle types, and changing flows, this variety—especially during peak times—creates big challenges.

RL models can tackle these differences by learning to manage unique intersection setups, like varying lanes or conflict points [Wiering, 2000]. However, models trained in simple, low-traffic cases often fail in busy, complex ones. Adaptive RL methods, like meta-reinforcement learning, help agents adjust fast to new traffic levels by building on past lessons [Lou et al., 2022]. For example, meta-RL has been used in traffic signal control to adapt to demand shifts, showing potential for unsignalized settings [Zang et al., 2020c].

While meta-RL facilitates faster policy adaptation, it faces scalability challenges in large urban traffic networks. Treating each intersection or CAV as an agent could mean thousands of agents, each needing its own policy. Storing all these policies takes a lot of memory. A practical fix is using a deep neural network (DNN) with smaller, shared policies, trading some accuracy for efficiency [Finn et al., 2017]. Also, updating a meta-policy one agent at a time slows training in large systems. Research on multi-agent traffic control suggests parallel updates could help, but this

needs more testing in unsignalized networks [Chu et al., 2019].

In short, managing large unsignalized networks is hard due to diverse layouts and flows. More work is needed to build RL models that scale up and handle real-world intersection challenges effectively.

1.1.4 Conclusion

Optimizing large-scale networks of unsignalized intersections is a challenging task due to the complexity of urban traffic systems. While HRL and MARL offer promising solutions through scalable strategies, its application to large-scale unsignalized traffic networks remains under-explored. Key challenges include balancing the granularity of hierarchical task decomposition in the learning process, generalizing policy to dynamic traffic patterns, and managing the heterogeneity of intersection geometries. Further research is needed to address issues such as coordination among agents, non-stationarity, and the scalability of RL frameworks to ensure effective and efficient traffic management in realistic scenarios.

1.2 Research Objective and Questions

Arising from the analysis presented in the previous sections, the **overarching research question** is:

Can a hierarchical, multi-agent reinforcement learning approach effectively optimize traffic flow in large-scale heterogeneous unsignalized urban traffic networks?

Specifically, does this approach improve traffic flow in terms of throughput, travel time, and travel-time reliability compared to existing methods?

To answer this question, the thesis addresses the following key sub-questions, which guide the design, development, and evaluation of the proposed framework:

1. **What is the appropriate hierarchical structure for a decentralized traffic management framework?** This question explores the design of a cohesive hierarchical framework, including task decomposition into manageable sub-problems, and the selection of state representations, action spaces, and reward functions at each level. The aim is to ensure that the hierarchical layers balance computational efficiency, interaction granularity, and global optimization objectives while improving convergence stability in decentralized environments.
2. **How can representation of intersection geometries be standardized?** This question focuses on developing a standardized representation model for diverse intersection layouts and traffic conditions. It examines how such a model can support automated collision detection mechanisms and reduce computational complexity while remaining adaptable to various urban intersections without requiring manual calibration.
3. **How can the framework balance short-term individual goals with long-term traffic network objectives?** This question addresses the design of reward functions and policies that harmonize short-term objectives, like individual vehicle efficiency, with long-term goals,

such as overall traffic flow optimization. The goal is to mitigate conflicts between hierarchical layers and ensure consistency in optimization across the entire system.

4. **What are effective strategies for training hierarchical multi-agent systems for large-scale, heterogeneous urban networks?** This question investigates training methodologies for each hierarchical layer, such as leveraging multi-agent proximal policy optimization (MAPPO) [Yu et al., 2022] for high-level network-wide optimization and parallel neural Monte Carlo tree search (PNMCTS) [Shi et al., 2024a] for mid-level intersection scheduling. It also examines centralized learning approaches and coordinated policy optimization to enhance efficiency and scalability across diverse urban traffic patterns.

The proposed framework is evaluated based on its ability to adapt to dynamic traffic conditions and varying intersection geometries while ensuring collision-free operations and minimizing travel delays. Key performance metrics include travel-time reliability, average travel time, total throughput, and waiting time benchmarked against existing traffic management approaches.

1.3 Thesis Approach and Contributions

This thesis tackles key questions in urban traffic management by developing a hierarchical, decentralized MARL framework. The framework systematically addresses these challenges through a layered design, where each module targets a specific aspect of the problem, enhancing traffic flow, coordination, and safety for CAVs.

1. Hierarchical Structure for Decentralized Traffic Optimization The framework is structured into three layers: The *high-level layer* optimizes network-wide throughput and platoon length, maximizing local and global traffic demands. The *mid-level layer* schedules CAV platoon sequences at intersections to minimize delays and prevent conflicts. The *low-level layer* provides precise vehicle trajectory control, ensuring smooth, collision-free navigation.

Contribution: This layered HRL structure decomposes complex urban traffic management into scalable, decentralized decision-making components, enabling distributed coordination among intersections.

2. Standardized Intersection Representation with Collision Detection To manage diverse intersection geometries, we introduce a transformation model based on a discrete-time state-space model (DTSSM) [Zhang et al., 2020a]. This model abstracts intersections into a uniform and structured representation, enabling automated identification of conflict areas and real-time collision avoidance. By eliminating the need for manual adjustments, the approach improves adaptability across heterogeneous urban scenarios, allowing seamless integration of new intersection layouts.

Contribution: This transformation model provides a systematic and automated method for standardizing intersection geometries, allowing scalable deployment of MARL-based traffic optimization across diverse urban intersections.

3. Reward Design for Balancing Short- and Long-Term Objectives A structured reward function aligns short-term objectives, such as minimizing crossing delays, with long-term traffic flow

optimization. Each hierarchical layer incorporates reward signals that guide local decisions while ensuring global system efficiency. This approach mitigates conflicts between individual platoon performance and overall traffic throughput, supporting robust policy learning in dynamic environments.

Contribution: The proposed multi-layer reward design ensures effective coordination between local intersection scheduling and system-wide traffic flow, preventing misaligned agent objectives.

4. Effective Training Strategies for Hierarchical Multi-Agent Learning in Heterogeneous Urban Networks The framework employs decentralized MARL, where each intersection acts as an independent agent that adapts to local traffic conditions while coordinating with neighbouring intersections. Traffic pressure data informs dynamic throughput adjustments, preventing bottlenecks and improving network-wide efficiency. MAPPO-based centralized training and decentralized execution enables agents to generalize across varying traffic patterns, with zero-shot and few-shot learning techniques enhancing adaptability in unseen scenarios.

Contribution: This decentralized MARL strategy enables intersections to dynamically optimize throughput in response to real-time traffic variations, improving coordination and adaptability in large-scale urban networks.

1.4 Assumptions

The proposed framework for urban traffic management operates under the following assumptions at each level, ranging from low-level vehicle trajectory control to high-level throughput optimization. These assumptions ensure the feasibility of the system's implementation and its ability to achieve efficient, safe, and scalable traffic management.

1. Top-level CAV Adoption: This thesis assumes that all vehicles within the traffic network are CAVs. This assumption underpins the system's design and enables the application of the strategies, such as platooning, precise trajectory control, and coordinated intersection management. Full CAV adoption is a reasonable assumption for future traffic systems as it aligns with ongoing trends and projections in the adoption of automated vehicle technologies [Häfner et al., 2021].

Studies have shown that the benefits of full CAV systems, including significant reductions in travel time, fuel consumption, and traffic congestion, are maximized when all vehicles are capable of real-time communication and automated decision-making [Talebpour and Mahmassani, 2016]. Additionally, full CAV scenarios eliminate the uncertainties associated with mixed traffic conditions, where the coexistence of human-driven and autonomous vehicles can lead to suboptimal system performance and increased safety risks [Pan et al., 2024].

While full CAV adoption may not reflect current conditions, it provides a valuable baseline for understanding the potential of advanced traffic management systems in a future fully automated environment. Such an assumption is frequently employed in the literature to evaluate the upper bound of system performance and to guide the development of policies and infrastructure for transitioning to fully autonomous traffic networks [Almobayedh, 2019].

2. Reliable Communication: This thesis assumes that all CAVs communicate consistently with minimal latency, enabling real-time command execution and trajectory updates. This is a reasonable assumption as vehicle-to-everything (V2X) communication technologies are rapidly advancing, providing the low-latency, high-reliability infrastructure required for real-time vehicular communication [Wang et al., 2020b], e.g., experimental studies demonstrate that modern V2X technologies can achieve latency as low as 1 millisecond, making them suitable for high-frequency updates in dense traffic environments [He et al., 2019a].

3. Accurate Vehicle Data: Each CAV is assumed to have access to precise, real-time data on its state, including position, speed, and acceleration. Advances in sensor technology, such as LiDAR, radar, and GPS, along with V2X communication, make this assumption increasingly realistic [Jing et al., 2017, He et al., 2019a]. These systems have demonstrated high accuracy under controlled and field conditions, providing the level of detail required for effective CAV coordination [Paden et al., 2016].

4. Global Traffic Awareness: Each agent deployed at an intersection is assumed to have sufficient information about neighbouring traffic to make effective decisions. Edge computing and cooperative perception allow neighbouring agents to access aggregated traffic data in real time, enabling informed, coordinated actions [Wu et al., 2020, Liu and et al., 2020].

5. Environmental Conditions: It is assumed that environmental factors such as weather and road quality do not significantly degrade sensing or operational capabilities. This is supported by the continued development of robust, automotive-grade LiDAR/radar and sensor-fusion algorithms that sustain performance in rain, fog, and low light [Li and Ibanez-Guzman, 2020, Yeong et al., 2021].

6. CAV Low-Level Control and Tracking: Each CAV can accurately track its assigned trajectory in real time with bounded comfort limits (e.g., acceleration and jerk constraints).

7. Platooning and Intersection Control Policy: Before entering the intersection, vehicles operate under decentralized longitudinal control using the Intelligent Driver Model (IDM), which governs acceleration based on relative spacing and velocity to the leading vehicle [Treiber and Kesting, 2013]. Upon arrival at the intersection entrance, control is handed over to a centralized crossing scheduler that governs vehicle coordination through trajectory assignment. At this stage, vehicles are grouped into platoons only if they share the same movement direction (e.g., through, left-turn, or right-turn). Each platoon is treated as a single composite unit for scheduling, and a unified longitudinal motion profile is assigned to its members for safe crossing.

8. Pedestrians and Other Vulnerable Road Users (VRUs): This study deliberately excludes pedestrians, cyclists, and other VRUs from the intersection model to focus on the upper-bound performance of fully-automated, signal-free vehicular traffic. Such simplification is commonly used in autonomous intersection management (AIM) literature when throughput and travel time optimization are prioritized.

9. Safety Fallback and Degraded Modes: If the intersection scheduler cannot resolve vehicle conflicts within a design time bound, CAV platoons default to a conservative first-in-first-out policy. During this fallback, they adopt increased headways and reduced speeds until normal operation can

resume. This behaviour aligns with functional safety and graceful degradation principles advocated in AIM frameworks.

1.5 Thesis Roadmap

This thesis is structured into ten chapters, covering the background, related work, proposed methodologies, evaluation, and conclusions.

Chapter 2: Background. This chapter introduces key concepts, including Connected and Automated Vehicles (CAVs), Markov Decision Processes (MDPs), Reinforcement Learning (RL) algorithms (DQN, PPO, SAC), Multi-Agent RL (MARL), Hierarchical RL (HRL), and Monte Carlo Tree Search (MCTS). It also examines RL applications in real-time strategy (RTS) games and traffic simulation.

Chapter 3: Related Work. This chapter reviews existing research on traffic control systems. It covers the evolution of signalized control, the taxonomy of unsignalized traffic management, optimization architectures, evaluation metrics, and research gaps, such as large-scale multi-intersection scenarios and platooning strategies.

Chapter 4: Overall Design for the Hierarchical Multi-Agent Deep Reinforcement Learning Framework. This chapter presents a hierarchical RL framework with three levels: high-level throughput optimization via MAPPO and traffic pressure methods, mid-level scheduling using Parallel Neural Monte Carlo Tree Search (PNMCTS), and low-level vehicle trajectory simulated in SUMO.

Chapter 5: Optimizing Intersection Scheduling. This chapter develops a crossing scheduling model using an intersection transformation approach and a discrete-time state-space representation. It introduces PNMCTS for decision-making and applies curriculum training strategies.

Chapter 6: Network-wide Traffic Optimization. This chapter presents a comprehensive approach to high-level platoon coordination in heterogeneous traffic networks. It details the design of a geometry-agnostic learning framework, incorporating a sub-lane traffic pressure-based state representation, a multi-lane based action space, and carefully designed neighbouring reward functions. The training is based on the Multi-Agent Proximal Policy Optimization (MAPPO) algorithm, enabling scalable and efficient policy learning for network-wide autonomous traffic control.

Chapter 7: Implementation of Vehicle Trajectory Control in SUMO. This chapter addresses real-time vehicle speed synchronization and multi-segment acceleration control. It formalizes the problem, presents an optimization algorithm, and provides mathematical proofs for trajectory control.

Chapter 8: Evaluation. This chapter benchmarks the proposed methods against established baselines in both single- and multi-intersection traffic networks. It provides a systematic evaluation of intersection scheduling performance and network-wide traffic optimization, with a focus on large-scale scenarios based on the Dublin city network.

Chapter 9: Conclusion. This chapter summarizes the key technical contributions and addresses the research questions posed in the thesis. It also discusses the limitations of the current work and outlines promising directions for future research.

2 Background

The rapid development of information technology has opened new opportunities to address the growing challenges in traffic network management and intersection control. Effective management of large-scale unsignalized traffic networks, particularly in scenarios involving Connected and Automated Vehicles (CAVs), requires a deep understanding of advanced concepts, technologies, and methodologies. This chapter provides a detailed background on the key techniques and tools relevant to this thesis, focusing on their application to large-scale traffic management and intersection optimization. The background knowledge also lays the foundation for the presentation of relevant state-of-the-art research in the next chapter.

CAVs are expected to significantly transform modern transportation systems by improving traffic flow and optimizing intersection management through enhanced connectivity and autonomous decision making. However, managing large-scale traffic networks remains a challenging problem due to the non-stationary interactions between multiple agents and the incomplete or uncertain information available in such environments. In this context, modelling these uncertain dynamics effectively is crucial. The Markov Decision Process (MDP) provides a suitable framework for capturing the inherent uncertainty in traffic systems and modelling the sequential decision-making processes required for traffic optimization.

Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) have emerged as powerful approaches for solving MDP-based problems including traffic management. These techniques are particularly well-suited for dynamic and large-scale environments, enabling adaptive and efficient decision-making. Multi-agent Deep Reinforcement Learning (MADRL) extends these capabilities by allowing coordinated decisions among multiple agents, such as vehicles navigating a shared intersection, making it possible to optimize decentralized systems in a collaborative manner. Hierarchical Reinforcement Learning (HRL) further addresses the challenges of scalability and complexity by decomposing large decision-making problems into smaller, more manageable sub-tasks, making it more efficient to solve high-dimensional traffic-control scenarios.

In addition, Monte Carlo Tree Search (MCTS) and Neural Monte Carlo Tree Search (NMCTS) offer structured approaches for decision-making and planning in complex traffic environments, enabling effective exploration of possible actions and outcomes. These methods are particularly beneficial for handling large-action-space traffic optimization problems that require balancing exploration and exploitation over multiple time steps.

A key tool in this research is the Simulation of Urban Mobility (SUMO) traffic simulator, which

provides a flexible and detailed simulation environment that allows for the precise control of vehicle trajectories. This capability is critical for testing the effectiveness of RL-based traffic management strategies in large-scale and realistic traffic scenarios.

In the following sections, this chapter elaborates on these concepts in detail, presenting the necessary background for understanding the methodologies and technologies that underpin this research.

2.1 Connected and Automated Vehicles

CAVs are poised to revolutionize the transportation industry by improving vehicle coordination, mitigating the incidence of traffic accidents, and reducing congestion. According to research by the U.K. government [U.K. Government Service, 2016], even at low penetration rates of 25%, CAVs can significantly enhance traffic conditions. Specifically, a 12% reduction in delay, a 21% improvement in journey times, and an 77% improvement in journey time variability in urban scenarios (Table 2.1). These improvements underscore the transformative potential of CAVs in optimizing traffic systems.

Table 2.1: Urban model results for different CAV penetration rates [U.K. Government Service, 2016].

Scenario	Average delay (s)		Average journey time (s)		Journey time variability (s)	
	(s)	(%)	(s)	(%)	(s)	(%)
Base	65.91	–	277.78	–	88.38	–
25% CAV	57.70	12.4%	219.52	21.0%	19.74	77.7%
50% CAV	54.44	17.4%	205.35	26.1%	10.01	88.6%
75% CAV	51.89	21.3%	198.72	28.5%	7.24	91.8%
100% CAV	48.02	27.1%	192.64	30.7%	6.00	93.2%
Upper bound	46.36	29.7%	184.25	33.7%	5.71	93.5%

CAVs comprise two main sub-categories: connected vehicles (CVs) and automated vehicles (AVs). CVs rely on two-way wireless communication to enable real-time information sharing among components of the transportation system. CAV wireless communication technologies are broadly categorized as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X). V2V communication enables vehicles to exchange data on speed and position, enhancing safety by alerting drivers and other vehicles to potential hazards, thereby reducing accidents and congestion. V2I communication facilitates data exchange with infrastructure components, such as traffic signals, weather forecasts, and traffic conditions. For instance, intelligent traffic signals powered by V2I improve traffic flow and enable accurate travel time estimation [He et al., 2019b]. V2X communication combines V2V and V2I, expanding the information ecosystem and providing a safer and more informed driving environment.

Automated vehicles (AVs), on the other hand, perform some or all driving tasks without human intervention, thereby eliminating driver errors that often contribute to crashes, traffic flow oscillations, and shock waves [Suriyarachchi and Baras, 2021]. AVs are categorized based on their level of automation, as defined by the SAE International taxonomy [SAE International, 2018]. Currently, level 2 automation, which includes steering and acceleration control with an awareness of

surroundings, has been achieved and is exemplified by applications like Tesla Autopilot [Tatarek et al., 2017].

In this study, we rely on V2I and V2V communication to facilitate real-time information acquisition and vehicle guidance. Furthermore, we assume full automation of CAVs to evaluate the maximum performance of RL-based optimization algorithms without human intervention, focusing exclusively on the computational aspects of traffic system optimization. Additionally, we rely on the precise execution of CAV trajectories according to the planned paths, ensuring accurate implementation of control strategies.

2.2 Markov Decision Processes and Reinforcement Learning

The framework of MDPs [Puterman, 1994] provides a mathematical structure for decision-making under uncertainty.

An MDP is formally defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ is the set of states of the environment,
- $\mathcal{A}$ is the set of actions of the agent,
- $P(s'|s, a)$ is the transition probability function, which defines the probability of reaching state s' given the current state s and executing action a ,
- $R(s, a)$ is the reward function, specifying the immediate reward obtained after executing action a in state s ,
- $\gamma \in [0, 1]$ is the discount factor, determining the importance of future rewards.

Solving an MDP involves finding an optimal decision-making strategy, or policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, that maximizes the expected cumulative reward. RL provides a powerful approach to optimizing MDPs when transition probabilities $P(s'|s, a)$ and reward functions $R(s, a)$ are a priori unknown or difficult to model explicitly. Instead of relying on predefined models, RL agents learn optimal policies by interacting with the environment and adjusting their actions based on observed rewards.

The objective in RL is to learn an optimal policy π^* that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right].$$

By leveraging trial-and-error learning, RL algorithms approximate optimal policies through iterative updates, enabling effective decision making in complex and uncertain environments modelled as MDPs.

2.2.1 Value-Based Methods and Deep Q-Networks (DQN)

Value-based RL methods [Watkins and Dayan, 1992] focus on learning the optimal action-value function $Q^*(s, a)$, which represents the maximum expected return achievable from state s after taking action a following the optimal policy. The **Bellman** equation [Barron and Ishii, 1989] provides a recursive relationship for $Q^*(s, a)$:

$$Q^*(s, a) = \mathbb{E}_{s' \sim P(\cdot|s, a)} \left[R(s, a) + \gamma \max_{a'} Q^*(s', a') \right].$$

Deep Q-Networks (DQN) [Mnih et al., 2015] extend the Q-learning algorithm by using a neural network parametrized by θ to approximate the action-value function $Q(s, a; \theta)$. This approach enables the application of Q-learning to high-dimensional state and action spaces. The network parameters are optimized by minimizing the temporal difference (TD) error, defined as:

$$L(\theta) = \mathbb{E}_{(s, a, r, s')} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right],$$

where θ^- represents the parameters of a target network that is only periodically updated to stabilize training.

One significant limitation of DQN is **overestimation bias**, which arises due to the use of the max operator in the TD target:

$$y = r + \gamma \max_{a'} Q(s', a'; \theta^-).$$

The max operator amplifies noise in the Q -value estimates because it always selects the action with the highest predicted value, even if that prediction is inaccurate. This results in systematic upward bias in the expected maximum Q -value:

$$\mathbb{E} \left[\max_{a'} Q(s', a'; \theta^-) \right] \geq \max_{a'} Q^*(s', a'),$$

where $Q^*(s', a')$ represents the true action-value function. Overestimation bias propagates during training, inflating target values and leading to suboptimal policy learning.

To mitigate overestimation, several modifications to the DQN framework have been proposed. One notable improvement is the Double DQN algorithm [Van Hasselt et al., 2016], which decouples the action selection and evaluation processes to reduce the bias. In Double DQN, the target Q -value is computed as:

$$y = r + \gamma Q(s', \arg \max_{a'} Q(s', a'; \theta); \theta^-),$$

where the action $\arg \max_{a'} Q(s', a'; \theta)$ is selected using the online network, but the corresponding Q -value is evaluated using the target network. This modification significantly reduces the overestimation bias by preventing the max operator from directly acting on noisy Q -value estimates.

While DQN has demonstrated remarkable success in tasks such as Atari games [Mnih et al., 2013], its vulnerability to overestimation, combined with challenges in handling continuous action spaces and sample inefficiency, limits its applicability to more complex RL problems.

2.2.2 Policy-Based Methods and the Actor-Critic Framework

Policy-based methods [Sutton et al., 1999a, Silver et al., 2014] directly optimize the policy $\pi_\theta(a|s)$, parametrized by θ , using gradient ascent on the objective $J(\pi_\theta)$, where the objective function is describe as below:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) \hat{A}^\pi(s, a) \right],$$

where $\hat{A}^\pi(s, a)$ denotes the advantage function, quantifying how much better action a is compared to the average action in state s under the policy π_θ .

Although policy-based methods are well-suited for handling continuous action spaces, they often suffer from high variance in gradient estimates, leading to instability during training. Actor-Critic methods address this limitation by combining value-based and policy-based approaches. Specifically:

- The *actor* is the policy $\pi_\theta(a|s)$, which selects actions based on the current state.
- The *critic* estimates the value function $V^\pi(s)$, which evaluates the expected return of the policy.

The advantage function used to guide the policy update is computed as:

$$\hat{A}^\pi(s, a) = R(s, a) + \gamma V^\pi(s') - V^\pi(s),$$

where $R(s, a)$ is the reward received after taking action a in state s , and $V^\pi(s')$ and $V^\pi(s)$ are the estimated value functions for the successor and current states, respectively. This formulation reduces the variance in gradient estimates, stabilizing training and improving performance.

2.2.3 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) [Schulman et al., 2017b] is a state-of-the-art actor-critic algorithm designed to address instability in policy updates. PPO constrains updates by clipping the probability ratio between the old and new policies to ensure updates remain within a trust region.

The PPO objective is defined as:

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

where:

- $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between the new and old policies,
- $\hat{A}_t$ is the estimated advantage function that measures how much better an action a is compared to the expected value of actions under the current policy. Accurate estimation of $A^\pi(s, a)$ is crucial for stable and efficient policy optimization. The advantage function is formally defined as:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t),$$

where $Q^\pi(s_t, a_t)$ is the action-value function, and $V^\pi(s_t)$ is the state-value function.

- ϵ is a hyperparameter that controls the clipping range.

The clipping mechanism prevents overly large policy updates, which could destabilize training, while allowing sufficient flexibility for exploration. PPO also improves sample efficiency by reusing collected data for multiple mini-batch gradient updates, striking a balance between stability and computational efficiency. These features make PPO robust for handling continuous action spaces and high-dimensional problems.

However, estimating $Q^\pi(s_t, a_t)$ directly can introduce high variance, particularly in long-horizon environments. PPO mitigates this by leveraging Generalized Advantage Estimation (GAE) [Schulman et al., 2015], a variance-reduction technique that stabilizes policy gradients. GAE balances bias and variance by combining Monte Carlo and temporal difference (TD) learning. It computes the advantage function as a weighted sum of n -step TD residuals:

$$\hat{A}_t^\pi = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_t^{(l)},$$

where:

- $\delta_t^{(l)} = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$ is the TD residual,
- γ is the discount factor, and
- λ controls the trade-off between bias (small λ) and variance (large λ).

By tuning λ , GAE interpolates between one-step TD learning ($\lambda = 0$) and Monte Carlo estimation ($\lambda = 1$), smoothing advantage estimates and improving gradient stability. PPO adopts GAE to enhance sample efficiency and policy updates, making it particularly effective in environments with noisy or sparse rewards.

Why PPO is Well Suited for Unsignalized Traffic Management

In unsignalized traffic management, agents deployed on intersections must make dynamic and adaptive decisions based on interactions with multiple arriving vehicles. The choice of RL algorithm is crucial in ensuring robust and efficient traffic flow management. PPO is preferable over other policy-gradient algorithms for the following reasons:

- **Stability in Learning:** PPO's policy constraints ensure more conservative gradient using clipping and entropy loss techniques, making it easier to implement and tune. This simplicity makes PPO a more practical choice for large-scale deployment in unsignalized traffic management.
- **On-Policy Learning for Rapid Adaptation:** In dynamic traffic environments, the state distribution shifts frequently. PPO, being an on-policy method, adapts more effectively to such changes, as it continually updates policies using the latest observations rather than relying on outdated replay buffer data.
- **Reliable Policy Updates:** PPO's importance sampling technique ensures that policy updates

remain within a reliable range, avoiding excessive exploration that may lead to inefficient or unreliable decision making.

2.3 Multi-Agent Deep Reinforcement Learning

MARL extends the framework of RL to environments where multiple agents interact with each other and the environment to optimize their individual or collective objectives. Unlike single-agent RL, MARL presents additional challenges due to the non-stationarity introduced by the simultaneous learning and adaptation of multiple agents. This non-stationarity arises because each agent's policy changes over time, altering the environment dynamics from the perspective of other agents, which complicates the learning process [Papoudakis et al., 2019].

2.3.1 Key Concepts in MARL

In a typical MARL setting, the environment is represented as a multi-agent Markov Decision Process (MMDP), defined by the tuple:

$$\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i \in \mathcal{N}}, P, \{R_i\}_{i \in \mathcal{N}}, \gamma \rangle,$$

where:

- $\mathcal{N}$ is a set of agents, with $|\mathcal{N}| = N$,
- $\mathcal{S}$ is a set of global states,
- $\mathcal{A}_i$ is the action space of agent i ,
- $P(s'|s, \mathbf{a})$ is the transition probability function, where $\mathbf{a} = \{a_1, a_2, \dots, a_N\}$ represents the joint action of all agents,
- $R_i(s, \mathbf{a})$ is the reward function for agent i , which depends on the global state s and the joint action $\mathbf{a}$,
- $\gamma \in [0, 1]$ is the discount factor, controlling the weighting of future rewards.

Each agent i independently selects actions $a_i \in \mathcal{A}_i$ based on its policy $\pi_i(a_i|o_i)$, where o_i is the observation of agent i . The objective of MARL is to learn policies π_i that maximize each agent's expected cumulative reward:

$$J_i(\pi_i) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, \mathbf{a}_t) \right].$$

2.3.2 Challenges in MARL

MARL presents several unique challenges compared to single-agent RL:

- **Non-Stationarity:** Since the policies of other agents evolve during training, the environment becomes non-stationary from the perspective of any individual agent, complicating policy learning.

- **Partial Observability:** Agents typically have access only to partial observations of the global state, requiring them to infer the missing information.
- **Scalability:** The size of the joint action space grows exponentially with the number of agents, leading to increased computational complexity.
- **Coordination and Cooperation:** In cooperative settings, agents must learn to coordinate their actions to achieve shared objectives, which requires effective communication and collaboration mechanisms.

2.3.3 Training Frameworks for Multi-Agent Reinforcement Learning

To address these challenges, frameworks like Centralized Training with Decentralized Execution (CTDE) and advanced policy gradient algorithms such as Multi-Agent Proximal Policy Optimization (MAPPO) [Yu et al., 2022] have been developed.

Centralized Training with Decentralized Execution

CTDE is a widely adopted framework in MARL that allows agents to utilize global information during training while relying only on local observations during execution. This approach aims to balance the benefits of centralized coordination and the scalability of decentralized decision-making.

During training, agents have access to a shared global state, which helps in stabilizing the learning process and improving coordination among agents. However, during execution, each agent makes decisions independently based only on its local observations and learned policies. This paradigm is particularly effective in partially observable environments where direct communication between agents is limited or costly.

One of the foundational algorithms following this framework is Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [Lowe et al., 2017], which extends the Deep Deterministic Policy Gradient (DDPG) [Lillicrap et al., 2016] algorithm to multi-agent settings by incorporating a centralized critic with access to the global state, while each agent maintains its decentralized actor network for policy execution. This combination allows for better credit assignment and more stable training.

Multi-Agent Proximal Policy Optimization (MAPPO)

Multi-Agent Proximal Policy Optimization (MAPPO) [Yu et al., 2022] is a policy gradient algorithm specifically adapted for cooperative multi-agent environments. It is an extension of the PPO algorithm, tailored to address the challenges of MARL, particularly within the CTDE framework. By combining PPO's clipped surrogate objective with centralized value estimation, MAPPO offers improved stability, scalability, and sample efficiency in multi-agent settings.

Training with a Centralized Critic During training, MAPPO employs a centralized critic that has access to the full global state and the joint actions of all agents. This centralized critic evaluates the value function, $V(s)$, which helps each agent learn a more accurate estimate of future

rewards. Unlike decentralized value-based approaches, the centralized critic provides a richer and more stable learning signal by considering the joint state-action space.

The centralized critic allows agents to better coordinate their actions by learning the relationships between their individual behaviours and global performance. However, once the training phase is completed, MAPPO agents discard the centralized critic and execute their policies independently. Each agent relies solely on its local observations to make decisions, ensuring scalability and adaptability in real-world decentralized systems. This execution strategy makes MAPPO well-suited for cooperative tasks in partially observable environments, such as traffic control [Wei et al., 2019a], multi-robot coordination [Canese et al., 2021], and distributed resource management [Wang et al., 2019].

2.4 Hierarchical Reinforcement Learning

HRL improves the efficiency, scalability, and decision-making of RL by organizing tasks into hierarchically structured sub-tasks. Unlike standard RL, which treats each action as a single decision, HRL enables agents to operate at multiple temporal and spatial resolutions [Botvinick, 2012, Pires et al., 2023], making it suitable for long-horizon, high-dimensional problems.

2.4.1 Structure of HRL

The core idea behind HRL is to introduce a **hierarchical** structure where multiple policies operate at different levels:

- The **high-level policy** (also known as the *meta-controller* or *manager*) focuses on long-term decision-making. It defines sub-goals or options at a coarser level. The high-level policy decides when to switch or update these sub-goals based on the current state and long-term objectives.
- The **low-level policy** (also called the *controller*) executes actions to achieve the sub-goals set by the high-level policy. This policy typically operates at a finer granularity, focusing on short-term tasks within the current sub-goal context.

This hierarchical structure introduces **temporal abstraction**, where decisions at different levels are made on different timescales. For example, in robotic navigation, the high-level policy might select a target location, while the lower-level policy manages immediate actions such as obstacle avoidance and movement.

2.4.2 Advantages of HRL

HRL offers several significant advantages over standard RL approaches, which make it well-suited for complex real-world applications.

Scalability and Complexity Reduction

By breaking down a task into hierarchically structured sub-tasks, HRL reduces the complexity of learning in large state and action spaces. Instead of learning a policy that directly maps states to actions in a high-dimensional space, HRL enables policies to focus on simpler sub-problems, which can be learned more efficiently.

For example, in a robotic manipulation task [Gebauer et al., 2021], a high-level policy can focus on determining a sequence of sub-tasks (e.g., pick, move, place), while low-level policies manage the detailed motor force control required to execute each sub-task.

Temporal Abstraction and Long-Horizon Decision-Making

Temporal abstraction is one of the most important features of HRL. In standard RL, the agent must make a decision at every timestep, which can become impractical for tasks requiring long-horizon planning. HRL allows high-level policies to make decisions less frequently while delegating the detailed execution to lower-level policies. This reduces the overall decision-making complexity and makes it feasible to solve long-term tasks.

Transferability and Reusability of Sub-Tasks

A key advantage of HRL is the ability to transfer learned sub-tasks across different tasks or environments. Once a low-level policy for a specific sub-task (e.g., obstacle avoidance in navigation) is learned, it can be reused in other environments without retraining. This makes HRL highly efficient in multi-task and lifelong learning scenarios.

Improved Exploration

In standard RL, exploration in large state spaces can be inefficient and often leads to sub-optimal solutions. HRL improves exploration by allowing the high-level policy to guide the search towards meaningful sub-goals, reducing the likelihood of getting stuck in local optima. This structured exploration strategy is particularly useful in environments with sparse rewards.

Interpretability and Debugging

The hierarchical structure of HRL provides better interpretability compared to flat RL methods. Each sub-task can be associated with a clear objective, making it easier to understand the agent's decision-making process and diagnose potential issues. For instance, if an agent fails in a complex task, the failure can often be traced to a specific sub-task or a high-level decision, simplifying debugging and refinement.

2.5 Monte-Carlo Tree Search and Neural Monte-Carlo Tree Search

MCTS [Browne et al., 2012] is a popular algorithm for decision making in environments where the outcome of actions is uncertain or computationally expensive to evaluate. MCTS has been widely used in fields such as game AI, planning, and robotics. NMCTS is an extension of MCTS that

integrates neural networks to enhance the scalability and performance of the algorithm in complex environments. This section provides a detailed explanation of the workings of MCTS and NMCTS, highlighting their key components and functions.

2.5.1 Monte-Carlo Tree Search

MCTS is a sampling-based search algorithm that builds a decision tree incrementally by simulating outcomes from different actions. It balances exploration (sampling unexplored actions) and exploitation (focusing on promising actions) to progressively improve decision making. The algorithm operates iteratively, consisting of four main steps:

Steps in MCTS

1. **Selection:** Starting from the root node, the algorithm traverses the tree by selecting child nodes based on a selection policy. A common selection policy used in MCTS is the Upper Confidence Bound for Trees (UCT) policy [Garivier and Moulines, 2011], which balances exploration (visiting actions that are less explored) and exploitation (focusing on actions with high rewards). The UCT formula is defined as:

$$UCT = \frac{W_i}{N_i} + C \sqrt{\frac{\ln N}{N_i}},$$

where:

- W_i is the total reward accumulated for child i ,
- N_i is the visit count of child i ,
- N is the visit count of the parent node,
- C is a constant that controls the exploration-exploitation trade-off.

The UCT formula consists of two terms:

1. **Exploitation Term** ($\frac{W_i}{N_i}$): This term represents the average reward of action i based on the outcomes of previous visits. Higher values indicate actions that have historically led to better rewards, encouraging the algorithm to focus on these actions.
2. **Exploration Term** ($C \sqrt{\frac{\ln N}{N_i}}$): This term increases the value of actions that have been visited less frequently relative to other actions. Specifically:
 - The numerator, $\ln N$, grows logarithmically with the total number of visits to the parent node, ensuring that actions with fewer visits are prioritized as the search progresses.
 - The denominator, N_i , penalizes actions that have already been visited many times, reducing their likelihood of being selected.
 - The constant C controls the balance between exploration and exploitation. Larger values of C favour exploration, while smaller values favour exploitation.

The exploration term effectively ensures that under-explored actions, which might have high potential rewards but lack sufficient data, are revisited periodically. This mechanism prevents the

algorithm from prematurely committing to suboptimal actions based solely on limited evidence.

The UCT policy embodies the principle of the **upper confidence bound**, where actions are evaluated not only based on their average rewards ($\frac{W_i}{N_i}$) but also on an uncertainty term ($C\sqrt{\frac{\ln N}{N_i}}$) that accounts for the lack of data. By doing so, UCT encourages a balanced search, prioritizing promising actions while continuing to gather information about less explored regions of the decision space. This balance is crucial for improving decision making accuracy as the search progresses.

2. **Expansion:** When a leaf node is reached, the algorithm checks if the node represents a terminal state. If not, it expands the tree by adding one or more child nodes corresponding to possible actions from the current state.

3. **Simulation (Rollout):** From the newly added node, the algorithm performs a simulation to estimate the reward. This involves randomly sampling actions until a terminal state is reached or a predefined depth is achieved. The outcome of the simulation provides an estimate of the reward for the expanded node.

4. **Backpropagation:** The result of the simulation is propagated back up the tree, updating the values of the visited nodes. Specifically, the visit counts (N_i) and total rewards (W_i) are updated to reflect the new information.

Advantages of MCTS

MCTS offers several benefits:

- **Scalability:** MCTS can handle large decision spaces by focusing computational resources on promising areas of the search tree.
- **Flexibility:** The algorithm does not require explicit models of state transitions or rewards, making it applicable to a wide range of problems.
- **Exploration-Exploitation Balance:** The UCT formula ensures that the algorithm effectively balances exploring new actions and exploiting known high-reward actions.

However, MCTS has limitations in environments with high-dimensional state spaces or complex reward structures, where simulating outcomes accurately becomes computationally expensive.

2.5.2 Neural Monte-Carlo Tree Search

NMCTS [Silver et al., 2017] builds upon the MCTS framework by incorporating neural networks to improve the efficiency and accuracy of decision-making. NMCTS leverages the predictive capabilities of neural networks to guide the search process and replace random rollouts with more informed simulations.

Key Components of NMCTS

1. **Neural Network Integration:** A neural network, typically a deep neural network (DNN) [LeCun et al., 2015], is trained to predict the value of a state ($V(s)$) and the probabilities of

selecting each action ($P(a|s)$). Instead of relying on random rollouts during the simulation phase, NMCTS uses the predicted state value $V(s)$ as an estimate of the reward.

2. **Policy Guidance:** The action probabilities $P(a|s)$ predicted by the neural network are used to guide the selection and expansion steps. Actions with higher probabilities are more likely to be explored, focusing the search on promising regions of the decision space.

3. **Improved Backpropagation:** The value predictions $V(s)$ are propagated back up the tree during the backpropagation phase, providing more accurate updates compared to MCTS.

Workflow of NMCTS

NMCTS follows the fundamental steps of MCTS while integrating neural network predictions to enhance decision-making. The neural network, typically trained via self-play or supervised learning on expert demonstrations, provides both prior probabilities for action selection and state value estimates, refining the search process.

- **Selection:** NMCTS modifies the standard MCTS selection process by incorporating a policy network's prior probability distribution, $P(a|s)$, and a value network's state evaluation, $V(s)$. The modified UCT formula is:

$$\text{NMCTS-UCT}(s, a) = Q(s, a) + c \cdot P(a|s) \cdot \sqrt{\frac{N(s)}{N(s, a) + 1}}, \quad (2.1)$$

where $P(a|s)$ biases exploration towards promising actions suggested by the policy network, accelerating convergence by prioritizing high-value regions of the search space.

Calculation of $Q(s, a)$ Using $V(s)$

In NMCTS, the policy network not only provides prior probabilities $P(a|s)$ but also outputs the state value $V(s)$, an estimate of the long-term reward from state s . The value function is used to refine action-value estimates $Q(s, a)$ through an iterative update rule:

$$Q(s, a) \leftarrow Q(s, a) + \frac{V(s') - Q(s, a)}{N(s, a)}, \quad (2.2)$$

where s' is the successor state after taking action a . This update mechanism integrates search with learned priors, refining $Q(s, a)$ to approximate the expected return more accurately. By leveraging the neural network's ability to generalize across similar states, NMCTS enhances sample efficiency and decision quality.

- **Expansion:** Nodes are expanded using the policy network's predicted action probabilities, prioritizing high-probability actions over uniformly random expansion. This guided expansion accelerates the search in large action spaces.
- **Simulation:** Instead of relying on random rollouts, NMCTS uses the value network's prediction $V(s)$ as a surrogate for future rewards. This significantly reduces computational

overhead while improving value estimation accuracy.

- **Backpropagation:** During backpropagation, both search-derived rewards and neural network predictions contribute to updating the tree. The value network refines its estimates through bootstrapping, ensuring stable and reliable updates.

Training the Neural Network in NMCTS

The neural network in NMCTS is trained iteratively through RL, using self-play and search-guided learning to improve both the policy and value functions. This training process ensures that the network effectively balances exploration and exploitation while generalizing across similar states.

The training pipeline consists of three key components:

- **Search-guided Learning :** The policy network is trained to imitate the action selection behaviour of NMCTS. Specifically, after performing multiple search iterations via self-play, NMCTS generates a distribution over actions based on visit counts, denoted as $\pi_{MCTS}(a|s)$. The policy network $P(a|s)$ is updated by minimizing the Kullback-Leibler (KL) divergence between these distributions:

$$L_{\text{policy}} = \sum_a \pi_{MCTS}(a|s) \log \frac{\pi_{MCTS}(a|s)}{P(a|s)} \quad (2.3)$$

This step aligns the learned policy with the search-improved distribution, ensuring that future searches are more efficient by leveraging prior knowledge from the network.

- **Value Function Learning:** The value network $V(s)$ is trained to predict the long-term reward of a given state, helping NMCTS reduce reliance on random rollouts. The training target for $V(s)$ is the discounted cumulative reward obtained from NMCTS rollouts. The objective function minimizes the mean squared error (MSE) between the predicted and actual values:

$$L_{\text{value}} = \mathbb{E}_{s \sim D} \left[(V(s) - \hat{V}(s))^2 \right], \quad (2.4)$$

where $\hat{V}(s)$ is the Monte Carlo estimate of state value obtained from search results.

Self-play allows the agent to improve without requiring external supervision, as it continuously challenges itself to perform better. Over time, as the policy and value networks refine their predictions, NMCTS becomes more efficient, enabling better decision-making in large and complex environments.

2.5.3 Applications of MCTS and NMCTS

Both MCTS and NMCTS have been applied successfully in various domains:

- **Game AI:** MCTS powered the success of AlphaGo [Silver et al., 2017, Silver et al., 2018], while NMCTS was later used in AlphaZero to combine tree search with deep neural networks for games like Go, Chess, and Shogi.

- **Robotics:** NMCTS has been used for motion planning and decision-making in robotic systems operating in dynamic environments.[Eiffert et al., 2020, Li et al., 2021b]
- **Traffic Management:** MCTS and NMCTS are employed to optimize traffic signal control [Peng et al., 2023] and vehicle routing [Pairet et al., 2022] by evaluating potential actions over multiple steps and improving system-wide efficiency.

Integrating neural networks into MCTS enhances decision-making by improving exploration, value estimation, and search efficiency. The policy network guides search toward promising actions, reducing unnecessary exploration, while the value network replaces costly rollouts with accurate predictions, stabilizing learning and lowering computational overhead. By leveraging past searches, NMCTS accelerates convergence to optimal strategies more efficiently than traditional MCTS. This synergy of search and learning enables NMCTS to navigate complex, high-dimensional state-action spaces, making it ideal for planning in sparse-reward environments. Together, MCTS and NMCTS provide a scalable framework for sequential decision-making in domains such as game AI, robotics, and traffic management.

2.6 Traffic Simulation and Control Frameworks

Managing traffic in large-scale urban networks, particularly at unsignalized intersections, poses significant challenges due to the complex interactions between vehicles and the need for real-time coordination. To develop and evaluate MARL-based traffic control strategies, simulation tools are crucial for modelling realistic traffic scenarios. Among the most widely used platforms are SUMO (Simulation of Urban Mobility) and TRACI (Traffic Control Interface), which provide a robust environment for simulating and interacting with traffic networks.

2.6.1 Simulation of Urban Mobility

SUMO (Simulation of Urban Mobility) is an open-source, highly portable traffic simulation platform designed for the microscopic modelling of traffic flow [Behrisch et al., 2011]. It allows for detailed simulation of individual vehicle movements within complex urban networks, making it ideal for evaluating traffic control algorithms in dynamic environments. SUMO provides several key features that support large-scale, real-world traffic scenarios:

- **Microscopic Traffic Simulation:** Models individual vehicles with detailed behaviour, including acceleration, lane-changing, and car-following dynamics.
- **Multi-Modal Traffic Support:** Simulates various modes of transportation, including cars, buses, bicycles, and pedestrians, providing a comprehensive representation of urban traffic.
- **Flexible Network Configuration:** Allows users to import road networks from real-world data (e.g., OpenStreetMap) or create custom networks for specific use cases.
- **Scalability:** Capable of simulating large-scale traffic networks with thousands of vehicles in real-time.

In the context of unsignalized intersection management, SUMO provides a reliable environment for

testing coordination strategies among multiple agents. By simulating realistic traffic flow patterns, it enables researchers to study the effects of different control policies and optimize multi-agent decision-making.

2.6.2 Traffic Control Interface

TRACI is a Python-based interface that enables real-time communication between external applications and the SUMO simulation. It allows for fine-grained control and monitoring of traffic elements, making it an essential tool for implementing MARL-based traffic management systems.

Through TRACI, external programs can interact with SUMO to:

- **Control Vehicle Behaviour:** Adjust vehicle speeds, reroute traffic, and manage interactions at intersections.
- **Retrieve Real-Time Data:** Access detailed information on vehicle positions, speeds, waiting times, and lane occupancy.
- **Dynamic Traffic Control:** Implement adaptive traffic management strategies based on real-time data from the simulation.
- **Multi-Agent Integration:** Seamlessly connect with multi-agent frameworks to enable decentralized traffic control at unsignalized intersections.

TRACI plays a critical role in enabling multi-agent systems to interact with SUMO in real-time, allowing agents to observe traffic states, execute actions, and receive feedback in a simulated environment. This integration supports the development of advanced coordination strategies for managing unsignalized intersections in large-scale urban networks.

2.7 Summary

This chapter outlined the foundational concepts and methodologies relevant to multi-agent and hierarchical traffic management in large-scale urban networks. The concept of CAVs was introduced, emphasizing their potential to enhance traffic efficiency through real-time communication and automation. The MDP framework and its extension to RL were discussed, with a focus on advanced algorithms such as DQN, PPO, and MAPPO.

HRL was highlighted for its ability to address the scalability and complexity issues associated with long-horizon tasks by decomposing them into manageable sub-tasks. NMCTS was described as a powerful tool for decision-making in uncertain environments, offering a balance between exploration and exploitation.

The chapter concluded with an overview of the SUMO traffic simulation platform and the TRACI interface, which provide a practical environment for testing and deploying multi-agent traffic control strategies. These tools enable the integration of real-time multi-agent systems into urban traffic management, offering a scalable and flexible approach to optimizing traffic flow at unsignalized intersections.

In the following chapters, we build upon this foundation by exploring state-of-the-art research in large-scale urban traffic management, focusing on advanced methodologies and their practical applications.

3 State-of-the-art Review

The efficient management of traffic, particularly at unsignalized intersections, has been an active area of research due to its potential to improve urban mobility and reduce congestion. This chapter provides an overview of related work on traffic management systems, beginning with a brief history of traffic-light control systems and progressing to unsignalized approaches. Signalized traffic control systems are examined to highlight their evolution and limitations, establishing the potential for alternative unsignalized strategies in the presence of connected and automated vehicles (CAVs). Following this, unsignalized traffic control systems are reviewed in detail, categorized by modelling approaches, optimization architectures, and intersection management strategies. Special attention is given to heuristic-based and reinforcement learning (RL)-based methods, as they tackle the dynamic and complex nature of urban traffic systems. The chapter concludes by highlighting gaps in existing approaches and emphasizing the development of scalable and adaptive frameworks as a promising direction to address the challenges of managing traffic in large-scale, unsignalized networks.

3.1 Evolution of Signalized Traffic Control Systems

Signalized traffic control systems have played a key role in managing urban traffic, improving alongside advancements in technology and the increasing challenges of traffic flow. This section gives an overview of how traffic light control systems have developed, starting from basic methods to more adaptive and smart systems. It also looks at advanced approaches that use learning-based techniques. Since this thesis focuses on managing traffic without signals, it is important to first understand how signalized systems work, how they have evolved, and what their limitations are. This will set the stage for discussing unsignalized traffic management methods.

3.1.1 Deployed Traffic Control Systems

The evolution of traffic signal control systems has seen significant advancements, starting with the introduction of fixed-time control systems. These systems operate based on predefined schedules and are effective under stable and predictable traffic conditions. However, their rigidity prevents them from adapting to real-time traffic fluctuations, resulting in substantial inefficiencies during unexpected traffic events such as congestion caused by accidents or sudden surges in demand [Tang et al., 2019].

To address these limitations, adaptive traffic control systems were introduced. One of the earliest and most notable examples is the *Split Cycle Offset Optimization Technique* (SCOOT), which

leverages real-time data from traffic sensors to dynamically adjust signal timings, thereby improving traffic flow and reducing delays. Studies show that SCOOT can enhance traffic performance by approximately 15% compared to fixed-time systems [Hunt et al., 1982]. Despite its advantages, SCOOT requires extensive sensor infrastructure and involves high setup costs, which limits its scalability and makes it less feasible for smaller cities or regions with constrained budgets [Stevanovic et al., 2010].

Another widely adopted adaptive traffic control system is the *Sydney Coordinated Adaptive Traffic System* (SCATS). SCATS adjusts signal phases based on real-time vehicle detection, optimizing traffic flow across intersections. While SCATS is effective in many scenarios, it faces challenges similar to those of SCOOT, including a strong dependency on comprehensive sensor networks. Furthermore, SCATS requires careful calibration and tuning for optimal performance, which can be labour-intensive and demands expertise. [Sims and Dobinson, 1980].

3.1.2 Advanced Traffic Control Systems

In recent years, reinforcement learning (RL) has shown great promise as an alternative to traditional methods for controlling traffic lights. One of the first works in this area was *PressLight* [Wei et al., 2019b], which builds on the concept of pressure in traffic control, where the pressure at an intersection is defined as the difference between the number of vehicles approaching and leaving it. By prioritizing phases with high pressure, *PressLight* aims to dynamically reduce congestion and balance the load across the network. [Wei et al., 2019b].

Subsequent research shifted focus from isolated decision-making toward coordination among multiple intersections. For example, *CoLight* [Wei et al., 2019d] employs graph attention networks to allow intersections to share information. This method enables each agent to consider both its local traffic data and the conditions at neighbouring intersections, reducing network-wide average travel time. Building on these ideas, further methods have focused on improving adaptability and transferability. For instance, *MetaLight* [Zang et al., 2020b] employs meta-reinforcement learning to allow policies to quickly adjust to new traffic conditions. In addition, *TransferLight* [Schmidt et al., 2024] shows how a policy learned in one network can be directly applied to another without retraining. Another approach, called *RELIGHT* [Qi et al., 2022], combines multiple explicit safety rules in RL models to prevent red-light running and aggressive left-turning behaviours.

More recently, another work—*CosLight* [Ruan et al., 2024]—addresses the limitations of existing RL-based traffic signal control methods, which often focus solely on neighbouring intersections. The authors identify that non-neighbouring intersections can also significantly impact traffic congestion, a factor often overlooked in traditional models. To tackle this issue, the authors propose a framework that decouples collaborator selection from the signal control policy, treating it as a separate policy to be learned concurrently. This selection policy dynamically identifies the most effective collaborators in real-time by analysing both phase-level and intersection-level features. By doing so, the system can adaptively choose the best set of intersections to collaborate with, beyond just immediate Neighbors.

Some other hierarchical methods are developed, where high-level tasks include traffic signal

optimization, while lower-level tasks target vehicle-level or platoon-level optimizations, each addressing different aspects to achieve efficient traffic flow and reduced congestion. These HRL-based traffic light control methods exhibit improved performance over deployed signal control systems [Borges et al., 2021, Xu et al., 2021a, Abdoos and Bazzan, 2021, Chen et al., 2018, Tan et al., 2020, taheer et al., 2020, Naveed et al., 2021, Zeng et al., 2022, Liu et al., 2023, Sun et al., 2023, Su et al., 2023]

3.1.3 Summary

The evolution of traffic signal control systems has advanced from fixed-time schedules to adaptive and learning-based methods. Fixed-time systems were simple but rigid, while adaptive systems like SCOOT and SCATS introduced real-time adjustments using sensor data, improving traffic flow but requiring high infrastructure costs. Recent reinforcement learning approaches, such as PressLight, CoLight, and CoSLight, focus on dynamic congestion management, multi-intersection coordination, and adaptive collaboration, offering better efficiency and scalability in modern traffic networks. This thesis focuses on unsignalized large-scale network control, which rely heavily on full-level vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, enabling real-time coordination among vehicles and intersections to enhance safety and reduce delays.

3.2 Taxonomy of Unsignalized Traffic Control Systems

Unsignalized traffic control systems have emerged as a promising solution for managing intersections without relying on traditional traffic signals, offering potential advantages in scalability, adaptability, and efficiency. This section presents a comprehensive taxonomy of unsignalized traffic control systems, categorizing them based on their modelling approaches, management architectures, and optimization strategies. It begins with an overview of modelling methodologies, including grid-based, collision-based, and vehicle-based approaches, which provide the foundation for understanding traffic dynamics at unsignalized intersections. The discussion then shifts to architectural frameworks for optimization, encompassing centralized, distributed, and hierarchical architectures, each with distinct operational paradigms. Finally, the section explores methods for optimizing traffic at both individual and multiple unsignalized intersections, highlighting heuristic-based and reinforcement learning (RL)-based strategies. This taxonomy offers a structured understanding of the field and serves as a foundation for identifying the strengths and limitations of existing approaches that rely on full automation of connected and automated vehicles (CAVs).

3.2.1 Modelling Unsignalized Intersections

To avoid potential conflicts between CAVs at an unsignalised intersection, prior research has generally adopted one of three primary modelling approaches: (1) Grid-based: leveraging a spatio-temporal occupancy map of the intersection area, (2) collision-point (area) based: using the prefixed trajectories of CAVs within the intersection, and (3) Vehicle-based: allowing complete route freedom for CAVs inside intersections.

Grid-based Modelling

The grid-based methodology models intersections as spatio-temporal grids of potential conflict zones, tracking each grid cell's occupancy by connected and autonomous vehicles (CAVs) over time [Bashiri et al., 2018, Dai et al., 2016, Wu et al., 2019b, Xu et al., 2020]. Specifically, a CAV's trajectory through the intersection is represented by identifying which cells within the grid will be occupied at each discrete time step

[Andert et al., 2017a, Aoki and Rajkumar, 2018, Azimi et al., 2013, Azimi et al., 2014]. To ensure collision avoidance, the intersection management system schedules CAVs such that no two occupy the same grid cell simultaneously. The granularity of grid partitioning varies across different approaches. In the simplest case, the entire intersection is treated as a single conflict area, referred to as intersection-based (IB) control [Dresner and Stone, 2004b, Dresner and Stone, 2008, Bashiri et al., 2018, Belkhouche, 2019, Bian et al., 2019, Zhang et al., 2016], where only one vehicle is allowed within the intersection at any given time, regardless of its route.

While the grid-based method is simple and efficient, it has some important downsides. The large grid cells can lead to very cautious control methods, allowing only a few vehicles to pass at the same time. This reduces the number of vehicles that can move through the intersection and can cause deadlocks, especially when traffic is heavy. Also, the fixed grid layout might not use the available space well, making it less effective for intersections with irregular geometries and changing traffic patterns.

Collision-point-based Modelling

The collision-point based approach, in contrast, does not require maintaining a detailed occupancy map of the intersection. Instead, it relies on the precomputed or predicted paths of the CAVs to identify potential conflict points [Kamal et al., 2015, Lu and Kim, 2016, Katriniok et al., 2018, Khayatian et al., 2018a, Katriniok et al., 2019]. Given that the expected paths of CAVs are typically predefined, such as paths for going straight, turning left, or turning right, it is feasible to determine the points of intersection between these paths offline. This significantly reduces the need for dynamic computation of occupancy, focusing instead on proactively scheduling manoeuvres to avoid collisions based on the expected trajectories. Thus, this approach is more computationally efficient but less adaptive to dynamic route changes.

Vehicle-based Modelling

Finally, the vehicle-based approach gives vehicles complete freedom of movement within the intersection [Li et al., 2018, Mirheli et al., 2019, Bian et al., 2020, Ge et al., 2022, Antonio and Maria-Dolores, 2022, Zheng et al., 2022]. In this scenario, vehicles have the autonomy to choose their optimal route to reach their desired exit lane. This methodology provides the highest level of flexibility and maximizes the use of the intersection area but poses a significant challenge in terms of computational requirements. Specifically, managing free movement within the intersection involves solving a multiple-vehicle scheduling problem of high complexity, requiring substantial computational power and advanced algorithms for effective conflict detection and resolution. Additionally, in real-world intersection sizes, it may not be feasible to allow many CAVs

to enter simultaneously to perform such manoeuvres.

Summary

Methodology	Advantages	Limitations	References
Grid-based	Simplifies collision avoidance by ensuring no two CAVs occupy the same cell; can vary in grid granularity	Limited capacity reduces throughput; computationally simpler but restrictive with only one vehicle allowed at times	[Bashiri et al., 2018, Dai et al., 2016, Wu et al., 2019b, Xu et al., 2020, Dresner and Stone, 2004b, Dresner and Stone, 2008, Bashiri et al., 2018, Belkhouche, 2019, Bian et al., 2019, Zhang et al., 2016]
Collision-point based	Reduces computational load by predefining conflict points and managing paths based on expected trajectories	Less adaptive to dynamic route changes; limited flexibility compared to vehicle-based models	[Kamal et al., 2015, Lu and Kim, 2016, Katriniok et al., 2018, Khayatian et al., 2018a, Katriniok et al., 2019]
Vehicle-based	Provides flexibility and maximizes intersection usage; allows dynamic path choice for vehicles	High computational demands due to free movement, requires advanced algorithms for conflict resolution	[Li et al., 2018, Mirheli et al., 2019, Bian et al., 2020, Ge et al., 2022, Antonio and Maria-Dolores, 2022, Zheng et al., 2022]

Table 3.1: Summary of CAV Conflict Modelling Methods

These three approaches differ in their trade-offs between computational complexity and the level of detail required for conflict detection. The grid-based approach employs conflict detection with low computational overhead, primarily due to the coarse granularity of the grid. However, this approach limits the number of vehicles allowed to enter due to coarse grid cells, and the fixed grid layout might not suit irregular intersections. The trajectory-based approach provides a more computationally efficient solution by leveraging prior knowledge of intersection geometry to predict CAV trajectories. In contrast, the vehicle-based approach allows for maximum flexibility and adaptability but comes with a significant computational overhead and complexity in conflict resolution in real intersections.

In the proposed framework of this thesis, collision-point-based modelling is applied as it provides a balanced solution that combines computational efficiency with the flexibility to handle irregular intersection geometries. This method focuses on specific collision points within the intersection, which reduces computational overhead compared to vehicle-based approaches that require continuous monitoring of each vehicle's trajectory. Unlike grid-based methods, which often suffer from coarse granularity and inefficiencies in irregular intersection layouts, collision-point-based modelling offers a more adaptable framework. It enables precise conflict detection by abstracting vehicle routes from the intersection geometry, making it suitable for managing complex traffic scenarios with diverse intersection designs.

3.2.2 Architectures for Managing Vehicle Crossing at Unsignalized Intersections

The fundamental distinction between centralised and decentralised strategies for intersection management lies in their dependency on the source of information: centralised approaches employ comprehensive global data from all CAVs within/approaching close to an intersection, while decentralised systems rely on local information accessible within the detection and vehicle-to-vehicle

(V2V) communication ranges of individual CAVs [Zhou, 2022].

Centralized Architecture

In a centralized traffic management framework, the Intersection Manager (IM) plays a pivotal role in coordinating CAVs at intersections [Rios-Torres and Malikopoulos, 2016]. This centralized approach is implemented primarily through reservation-based or analyst-based systems to regulate traffic flow [Wu and Qu, 2022].

Reservation-based systems are traffic management frameworks designed to coordinate vehicle entry, exit, and longitudinal control at intersections. These systems operate by enforcing strict adherence to predefined timing schedules, ensuring that each vehicle follows its assigned trajectory to avoid conflicts. One of the most notable reservation-based systems is the Automated Intersection Management (AIM) protocol, first introduced by Dresner and Stone in 2004 [Dresner and Stone, 2004a] and subsequently enhanced in 2008 [Dresner and Stone, 2008]. AIM operates using a First Come First Served (FCFS) policy, where requests to cross the intersection are processed in the order in which they are received. To facilitate vehicle coordination, AIM maintains a reservation grid that segments the intersection area, ensuring no two vehicles occupy the same segment concurrently. This FCFS strategy has been widely adopted in various other intersection management techniques as well [Elhadeif, 2015, Andert et al., 2017b, Qiao et al., 2018a, Khayatian et al., 2018b].

When traffic demand increases, reservation-based approaches face significant challenges, particularly in managing reservation competition, which can lead to throughput bottlenecks. Situations arise when multiple CAVs attempt to reserve overlapping collision points simultaneously. To address this issue, [Gregoire et al., 2014] proposed a graph-based priority mechanism to eliminate competition, while [Lin et al., 2022] introduced the idea of limiting vehicle throughput by imposing a maximum crossing threshold.

To address the challenges of throughput bottlenecks at intersections, a range of analyst-based methods have been proposed, evolving over time to improve traffic management. Early approaches focused on optimization models like Mixed Integer Linear Programming (MILP), which aimed to increase efficiency by dynamically forming vehicle batches and determining optimal service sequences under varying traffic conditions [Yu et al., 2019]. Although effective in structured traffic scenarios, these methods often struggled with scalability and adaptability to highly dynamic environments.

Building on this foundation, multi-agent systems were introduced, incorporating blockchain mechanisms to enhance transparency and coordination. In these systems, vehicles arriving at the intersection propose a crossing time slot, which is recorded and validated on a blockchain ledger. This ensures that the coordination process is tamper-proof and conflict-free, enabling multiple vehicles to cross safely and efficiently [Buzachis et al., 2020]. Despite these advances, the complexity of multi-agent coordination in real-time remained a significant delay.

To overcome trajectory conflicts and further enhance computational speed, heuristic-based methods such as tabu search and clustering algorithms were introduced. These techniques reduced trajectory overlap between vehicles, increasing the number of successful crossings and improving intersection

capacity [Su et al., 2020]. However, these methods rely on fixed trajectory assumptions and predefined speed profiles, which limit their applicability in highly dynamic and unpredictable traffic environments.

The next step in this evolution introduced more flexible traffic control strategies, such as lane-free AIM. This approach allowed vehicles to make dynamic velocity and path adjustments, leveraging batch processing to enhance crossing efficiency and promote cooperative driving behaviour [Li et al., 2021a]. Building on this, the concept of "all-direction" lanes was proposed to further optimize entry timing and route choices, significantly improving intersection performance [Wu et al., 2022].

To manage increasingly dynamic and complex traffic patterns, advanced RL techniques have been applied to intersection management. The Advantage Actor-Critic (A2C) algorithm, which combines policy-based and value-based learning, has been utilized to dynamically control the vehicle speed profiles, which can increase throughput and reduce waiting times, particularly in high-traffic conditions [Ayeelyan et al., 2022].

In summary, centralized control strategies in traffic management offer significant benefits but also pose several challenges that must be addressed.

Benefits:

- **Enhanced Coordination:** Centralized systems provide a unified platform for monitoring and controlling traffic signals and intersections, enabling real-time adjustments and maintaining stable performance as the network size increases.
- **Efficient Resource Allocation:** These systems optimize traffic management by dynamically controlling traffic signals and road resources on a global scale, reducing congestion and improving overall efficiency.
- **Scalability:** Centralized control allows the same policy to be deployed across the entire network, ensuring consistent behaviour and performance even as the traffic network grows, without requiring separate policies for different regions.

Challenges:

- **Data Dependency:** Centralized control relies heavily on prior knowledge and continuous data collection, which can be problematic in areas with limited or unreliable data infrastructure.
- **Processing Bottlenecks:** Simultaneously handling multiple requests can cause significant delays, especially during high-demand periods, reducing system responsiveness.

Distributed Architecture

Recent advancements in decentralised intersection management for CAVs have aimed to address the challenges of reducing congestion, enhancing throughput, and crossing safety. Multiple strategies have emerged, leveraging the capabilities of distributed decision-making and local observation and communication.

Framework	Approach	Objective	Metrics
[Dresner and Stone, 2004a, Dresner and Stone, 2008]	Reservation-based AIM (FCFS)	Coordinate vehicle entry/exit via a reservation grid that assigns exclusive time-space slots	Intersection throughput, travel delay
[Elhadeif, 2015]	FCFS-based AIM Variant with AI Enhancements	Extend the basic FCFS strategy by incorporating intelligent conflict resolution	Throughput, collision rate
[Andert et al., 2017b]	Crossroads AIM Extension	Adapt the FCFS approach for improved coordination in multi-intersection settings	Travel delay
[Qiao et al., 2018a]	Virtual Reservation Protocol	Use virtual reservations to manage conflicting collision points	Collision rate
[Khayatian et al., 2018b]	Robust Intersection Management (RIM)	Enhance the robustness and safety of reservation-based systems under variable traffic conditions	Collision rate, throughput consistency
[Gregoire et al., 2014]	Graph-based Priority Mechanism	Eliminate reservation competition by prioritizing conflicting requests using graph analysis	Deadlock frequency, collision rate
[Lin et al., 2022]	Maximum Throughput Threshold Approach	Limit vehicle throughput by imposing a crossing threshold to avoid deadlocks	Deadlock frequency, throughput
[Yu et al., 2019]	MILP for Batch Processing	Dynamically form vehicle batches and determine optimal service sequences under varying demand	Batch efficiency, travel time
[Buzachis et al., 2020]	Blockchain-based Coordination	Record and validate proposed crossing slots on a tamper-proof ledger	Transaction latency, collision rate
[Su et al., 2020]	Tabu Search and Clustering Algorithms	Reduce trajectory conflicts by clustering similar paths to minimize overlaps	Conflict rate
[Li et al., 2021a]	Lane-Free AIM with Batch Processing	Enable dynamic velocity and path adjustments by eliminating fixed lane constraints	Average wait time
[Wu et al., 2022]	All-Direction Lanes Strategy	Optimize entry timing and routing choices by allowing vehicles to use all available lanes	Average delay
[Ayeelyan et al., 2022]	RL-based Advantage Actor-Critic (A2C)	Dynamically control vehicle speed profiles to enhance throughput and ensure fairness in high-traffic conditions	Average/max wait time, throughput, fairness

Table 3.2: Summary of Centralized Intersection Management Frameworks

Auction-based and bidding-based decentralised intersection management mechanisms have received significant attention as promising solutions for resolving right-of-way conflicts at intersections. In these systems, vehicles act as autonomous agents that submit bids for priority, with criteria such as speed, proximity, and urgency influencing their bids. The allocation of crossing rights is typically handled by distributed negotiation protocols that aim to optimise overall traffic flow, reduce congestion, and enhance safety.

The development of auction-based systems has progressed from basic bidding schemes to more sophisticated frameworks that address both efficiency and fairness. [Morrisett et al., 2022] introduced an auction-based intersection management system where vehicles bid for feasible crossing times, primarily aiming to maximise throughput and space utility. Building on these foundations, [Li et al., 2023b] extended the concept by proposing a cooperative traffic management

framework using collection auctions to improve fairness. This advanced model incorporates driver preferences and historical behaviour into the bidding process. By balancing priority allocation and fairness, the framework ensures more equitable access while aligning with broader traffic management objectives, including reducing congestion and improving travel time reliability.

While auction-based intersection management systems offer flexibility and adaptability, they present specific challenges. One significant concern is the potential for drivers with more 'money' to consistently outbid others, leading to priority access based on financial capability, which undermines fairness [Santos et al., 2020]. Additionally, drivers might inflate bids without genuine urgency to gain priority, distorting the system's intended operation [Carlino et al., 2013]. Furthermore, in environments with both autonomous and human-driven vehicles, ensuring seamless participation in auction-based systems is complex, as human drivers may lack the technology for real-time bidding [Muzzini and Montangero, 2024].

To overcome the limitations of auction-based systems, researchers have developed precise and deterministic approaches using mathematical models to ensure consistent and reproducible outcomes. One prominent approach is based on Distributed Constraint Optimization Problems (DCOPs), where each vehicle functions as an autonomous agent, negotiating with others to resolve conflicts and coordinate intersection movements. This decentralized framework eliminates the need for centralized control and enhances scalability in dynamic traffic environments [Vu et al., 2018]. However, traditional DCOP formulations assume discrete variables and tabular constraint utilities, which do not align well with the continuous nature of real-world traffic, leading to modelling inaccuracies and computational inefficiencies [Fioretto et al., 2018].

Building on these challenges, [Iwase et al., 2022] introduced a significant improvement with a local interaction model. In this approach, each intersection functions as an autonomous agent, communicating only with neighbouring intersections, experiments are conducted on a urban network with 24 intersections. To enhance scalability, the algorithm replaces traditional tabular utility functions with a polynomial-time iterative conflict resolution mechanism, which adjusts vehicle time slots iteratively until a stable and feasible solution is reached. Despite these improvements, the NP-hard nature of coordinating multiple intersections present challenges for large-scale networks, limiting the algorithm's ability to guarantee globally optimal solutions in highly dynamic environments.

Additionally, harmony matrix-based coordination systems have been proposed. This approach enables autonomous vehicles to synchronize their movements through intersections without relying on traffic signals, allowing multiple vehicles to pass simultaneously and enhancing both safety and throughput [Katole and Sinha, 2023]. However, implementing this system requires precise and heavy vehicle-to-vehicle communication and coordination. Any communication delays or failures can disrupt the synchronization process, leading to potential safety risks and reduced efficiency. To enhance information exchange and utilization, [Abbas-Turki et al., 2023] employed distributed particle swarm optimization (DPSO) to optimize traffic flow by enabling self-organized vehicles to adapt to traffic demand while efficiently utilizing collision points. [Ali et al., 2020] apply Non-linear MPC with a quadratic cost equation to manage the vehicle dynamic to avoid collisions.

Recently, Reinforcement Learning (RL) has been increasingly applied to address the challenges of decentralised unsignalized intersection management. For instance, [Qiao et al., 2018b] implemented a Deep Reinforcement Learning (DRL) approach for controlling individual CAVs to safely navigate through intersections. Similarly, [Wang et al., 2020a] developed a DRL model that enabled a CAV to execute left turns at signal-free intersections while avoiding collisions. Although these studies demonstrate the feasibility of using RL for individual vehicle control at intersections, their primary focus remains on isolated decision-making for single CAVs.

To address the limitations associated with isolated CAV behaviour, [Wu et al., 2019b] proposed a decentralised RL-based framework known as DCL-AIM, which aims to coordinate multiple CAVs at unsignalised intersections. DCL-AIM formulates the CAV movement as a multi-agent Markov Decision Process (MDP), wherein each vehicle is represented by an RL agent responsible for optimising its acceleration. The objective of this approach is to reduce intersection delays while ensuring a collision-free environment. Despite promising results, the scalability of DCL-AIM is limited as it primarily focuses on coordinating only two interacting CAVs at a time, which significantly restricts its applicability in more complex traffic settings involving multiple agents.

To further enhance decentralised AIM, [Antonio, 2022] introduced RAIM (Reinforced AIM), a multi-agent control framework leveraging Long Short-Term Memory (LSTM) networks to regulate individual vehicle speeds. The RAIM model is capable of handling a complex, four-way three-lane intersections for multiple vehicles simultaneously, ensuring collision-free passage through real-time adaptation to dynamic conditions.

While decentralised approaches have led to significant progress, they still face notable challenges [Iliopoulou et al., 2022], particularly in terms of achieving global optimality, computation overhead, and scalability in large-scale traffic networks.

1) Difficulty in achieving optimal global coordination: In decentralised frameworks, each CAV operates independently, relying solely on local information and individual decision-making. This localised decision-making increases the risk of conflicting actions in larger traffic networks, especially under large-scale intersection geometries. Without a coordinated strategy, there is a heightened risk of deadlocks, increased overall delays, and potential safety hazards.

2) Computation overhead: Decentralised approaches often demand substantial computational resources to process local information, simulate traffic dynamics, and optimise control strategies in real time. This becomes more challenging as the number of agents increases, with interactions among vehicles causing a surge in the complexity of the computations. High computational requirements can lead to slower decision-making and reduced system responsiveness, making these methods less feasible for real-time traffic control in practical scenarios.

3) Scalability limitations: Decentralized approaches are often limited to managing individual intersections, focusing on local CAV dynamics. Scaling these systems to large urban networks with thousands of vehicles and interconnected intersections is challenging due to increased computational demands and the difficulty of ensuring effective coordination. These limitations reduce their ability to handle the complexity of real-world urban traffic.

Framework	Approach	Objective	Metrics
[Morrisett et al., 2022]	Auction-based AIM (Basic Bidding)	Enable vehicles to bid for feasible crossing times to maximize throughput and space utility	Waiting time
[Li et al., 2023b]	Auction-based AIM with Collection Auctions	Incorporate driver preferences and historical behaviour to balance priority allocation and fairness	Fairness, travel time
[Santos et al., 2020]	Auction-based Fairness Analysis	Examine potential equity issues where financial capability may dominate bid outcomes	Fairness, travel time
[Vu et al., 2018]	DCOP-based Intersection Management	Allow each vehicle (agent) to negotiate crossing slots and resolve conflicts in a decentralized manner	Waiting time
[Iwase et al., 2022]	Local Interaction Model with Iterative Conflict Resolution	Enhance scalability by enabling intersections to coordinate only with Neighbors via a polynomial-time conflict resolution mechanism	Travel delay
[Katole and Sinha, 2023]	Harmony Matrix-Based Coordination	Synchronize vehicle movements without traffic signals to allow simultaneous crossing	Throughput, collision rate
[Abbas-Turki et al., 2023]	Distributed Particle Swarm Optimization	Dynamically adjust vehicle speed	Throughput
[Ali et al., 2020]	Non-linear Model predictive Control	Dynamically adjust vehicle speed	Throughput
[Qiao et al., 2018b]	DRL for Individual CAV Control	Use deep reinforcement learning to enable safe navigation through intersections by individual CAVs	Crossing success rate, collision rate
[Wang et al., 2020a]	DRL for Left Turn Execution	Enable CAVs to execute left turns safely at signal-free intersections	Left-turn success rate, collision rate
[Wu et al., 2019b]	Decentralized RL-based AIM (DCL-AIM)	Coordinate multiple CAVs at unsignalized intersections to reduce delays and ensure collision-free movements	Average delay, collision rate
[Antonio, 2022]	RAIM with LSTM Networks	Regulate individual vehicle speeds in complex, multi-lane intersections for real-time, collision-free passage	Collision rate, throughput, travel delay

Table 3.3: Summary of Decentralized Intersection Management Frameworks

Hierarchical Architecture

Hierarchical decomposition in traffic control involves breaking down complex traffic management tasks into smaller, more manageable subproblems, where each level focuses on distinct optimization objectives. This layered approach allows traffic control systems to target specific components of the traffic network independently while contributing to overarching system goals. The hierarchical framework is inherently flexible, making it adaptable to a variety of optimization requirements and suitable for diverse applications in intelligent traffic systems.

Several studies have introduced hierarchical reinforcement learning (HRL) frameworks to enhance traffic efficiency through multi-level control. In the context of automated control for unsignalised intersections, one effective approach is to define the optimization of platoon crossing routes as a high-level objective, with individual vehicle manoeuvres managed at a low level. This hierarchical structure enables more coordinated traffic flows while maintaining detailed control over vehicle behaviours. For instance, [Al-Sharman et al., 2023] presents an innovative HRL framework designed specifically for automating left-turn manoeuvres at unsignalised intersections. This framework utilises the Soft Actor-Critic (SAC) algorithm to handle high-level route planning, effectively

mapping out optimal crossing paths. Model predictive control (MPC) is employed at the low level to manage real-time vehicle motion, ensuring that each left-turn manoeuvre aligns with dynamic traffic conditions. This dual-layered approach seeks to achieve accurate left-turn behaviours while incorporating considerations of vehicle dynamics, adherence to traffic regulations, and ride comfort, aiming to enhance both safety and fluidity in the intersection environment. [Hamouda et al., 2021] presented a multi-layer control architecture combining NMPC and Deep Deterministic Policy Gradient (DDPG) algorithm for unsignalised intersections, achieving a high success rate in collision avoidance. [Qiao et al., 2021] introduced a HRL-based behaviour planning structure for urban environments, outperforming heuristic-rule-based methods in left turns and lane changes. The Local Attention Safety Deep Reinforcement Learning (LA-SRL) framework is another decision-making approach designed to optimize vehicle kinematic strategies in dynamic, unsignalized intersection scenarios [Li et al., 2024]. It adapts the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm to handle continuous action spaces effectively, addressing the challenges of complex traffic environments. The framework incorporates a local attention mechanism that enhances situational awareness by focusing on critical interactions between the ego vehicle and surrounding traffic. This mechanism allows the model to dynamically prioritize nearby vehicles that significantly impact the ego vehicle's decisions, improving precision and computational efficiency. COOR-PLT [Li et al., 2023a] framework employs a two-tiered architecture, where the high-level planner dynamically forms platoons to optimize traffic flow, while the low-level policy executes fine-grained trajectory control to ensure safe and efficient navigation. The model demonstrates improved throughput, reduced travel time, and enhanced safety compared to rule-based and model-predictive approaches, making it a promising advancement in unsignalized urban traffic management. While innovative, COOR-PLT's reliance on DQN for decision-making lags behind more sample-efficient methods like PPO [Schulman et al., 2017a], potentially leading to slow convergence and overestimation issues in complex traffic networks. Moreover, although it advances the concept of adaptive platooning, it lacks experiments in handling heterogeneous intersections layouts, limiting its generalizability and practical applicability.

In conclusion, HRL represents a powerful approach for intersection management, offering scalability, adaptability, and superior traffic optimization capabilities. However, its effectiveness depends on careful framework design to ensure seamless coordination between hierarchical layers, robust training methods, and practical considerations for deployment in real-world traffic systems. Addressing these limitations while leveraging its strengths positions HRL as a vital tool for the future of intelligent traffic management.

3.2.3 Optimizing Crossing on Individual Intersections

This section explores the application of heuristic-based and reinforcement learning (RL)-based methods for optimising individual intersections. Heuristic approaches focus on rule-based and analyst-driven techniques to enhance scheduling efficiency, safety, and traffic flow. In contrast, RL-based methods leverage adaptive learning frameworks to address the dynamic and complex nature of intersection management, demonstrating potential in improving collision avoidance, traffic throughput, and adaptability in urban traffic systems.

Framework	Approach	Objective	Metrics
[Al-Sharman et al., 2023]	HRL with Soft Actor-Critic (SAC) for route planning and MPC for motion control	Optimize left-turn manoeuvres and enhance safety at unsignalized intersections	Collision rate, left-turn success rate
[Hamouda et al., 2021]	Multi-layer control with NMPC and DDPG for multi-agent coordination	Achieve high collision avoidance success rate at unsignalized intersections	Collision rate, safety metrics
[Qiao et al., 2021]	HRL-based behaviour planning for urban traffic	Outperform heuristic-rule-based methods in tasks like left turns and lane changes	Collision rate
[Li et al., 2024]	Local Attention Safety Deep RL (LA-SRL) with TD3 algorithm	Enhance situational awareness and optimize kinematic strategies at unsignalized intersections	Collision rate, delay, travel time
[Li et al., 2023a]	Deep Q learning	A hierarchical system design is employed to decompose the complex optimization problem into multiple layers	Throughput, travel time

Table 3.4: Summary of HRL Frameworks in Traffic Management

Heuristic-based Methods

The evolution of heuristic-based methods for isolated intersection management unfolds as a series of incremental improvements. Early work integrated a motion planning technique with the basic First-Come-First-Served (FCFS) rule to optimize vehicle arrival times and crossing speeds [Au and Stone, 2010]. Building on that foundation, later research enhanced the FCFS principle by clustering vehicles so that they could exchange information and adjust their travel speeds dynamically [Liang et al., 2018]. In a further refinement, subsequent studies incorporated conflict point analysis and explicitly accounted for wireless transmission delays by setting actuation times after the request plus the worst-case round-trip delay, with a PID controller determining the optimal velocity for intersection entry [Khayatian et al., 2019]. Together, these advancements progressively improved scheduling efficiency and safety within the FCFS framework.

Platoon-based methods have been extensively studied to enhance intersection performance by improving traffic flow efficiency and reducing delays. One of the early contributions in this domain introduced the BATCH algorithm [Tachet et al., 2016], which employs a fixed reordering period to aggregate incoming requests before determining velocities. This method significantly increased intersection capacity and reduced average delays by allowing better coordination of vehicle arrivals. [Bashiri and Fleming, 2017] proposed a platoon-based intersection management system that allows only one platoon in the conflict zone at a time. [Liang et al., 2018] introduced rule-based methods to refine platoon formation, particularly by splitting excessively long platoons into sub-platoons, thereby enhancing coordination and flexibility at intersections. Following these early developments, more sophisticated strategies have been proposed to further optimize platoon-based intersection management. [Bashiri et al., 2018] explored a reservation-based approach where vehicles adjust their speeds in real-time to form optimized platoons. [Bisht and Shet, 2020] extended this concept by incorporating V2X-based coordination strategies, allowing more responsive platoon management. Further refinements were introduced by [Kumaravel et al., 2021] proposed a two-layer optimization framework to determine the platoon size and crossing sequence, where the high-level job shop scheduler treats the intersection as a single machine and platoons as jobs. Similarly, [Mushtaq et al., 2021] proposed a two-level framework, where the high level employs swarm

optimization to form platoons, while the low level performs collision avoidance using V2V and V2I communication. Their results demonstrate that well-structured platoons can significantly enhance intersection throughput. Despite the advancements in platoon-based intersection management, most existing methods rely on static or pre-defined platoon formation rules, limiting their adaptability to real-time traffic variations and unexpected disruptions. While optimization-based approaches have been introduced to enhance flexibility, they often involve complex constraint formulations and high computational demands, posing challenges for real-time implementation in large-scale urban networks.

Virtual-lane-based strategies have also evolved to reduce conflicts and streamline vehicle movement. An initial approach projected traffic flows onto a one-dimensional virtual lane by representing conflicting movements with a directed graph and synchronizing vehicles via a depth-first search tree [Xu, 2018]. This idea was later extended by reconfiguring lane usage dynamically—temporarily allowing lanes normally reserved for the opposite direction to be used for uninterrupted flow thus further reducing conflicts [Mitrovic et al., 2020]. To address the undesired spillbacks at intersections, [Cong et al., 2022] and [Zhou et al., 2022] introduced virtual platooning strategies with feedback loops, where CAVs are projected onto a virtual lane based on conflict vehicle clusters, forming multiple virtual platoons to enhance safety and platooning accuracy.

Analytical methods offer a more sophisticated solution by framing intersection management as an optimization problem. An early effort formulated a centralized intersection controller using a mixed-integer linear program (MILP) to maximize the number of vehicles crossing while minimizing arrival time differences [Fayazi and Vahidi, 2017, Zhou et al., 2022]. This was followed by a joint control framework that integrated traffic signal control with trajectory optimization to reduce emissions and delay [Feng et al., 2018]. Later, a mixed-integer non-linear program (MINLP) was introduced for distributed vehicle trajectory planning, enabling vehicles to self-organize and secure safe, globally optimized trajectories [Mirheli et al., 2019]. Finally, dynamic programming techniques were applied to jointly optimize vehicle trajectories and signal controllers, further enhancing safety, reducing fuel consumption, and minimizing overall delay [Guo et al., 2019].

In summary, the progression from early FCFS enhancements to distributed protocols, platoon-based techniques, lane reconfiguration, and advanced analytical optimization demonstrates a robust evolution of heuristic-based methods. This build-up of innovations highlights the ongoing effort to balance efficiency, safety, and computational feasibility in real-world intersection management.

RL-based Methods

The evolution of reinforcement learning (RL) for intersection management can be traced through a series of studies that progressively build on earlier work. Initial efforts focused on incorporating vehicle-specific dynamic and kinematic constraints into deep RL models [Demir and Sezer, 2018]. This work established the importance of embedding inherent vehicle limitations into the decision-making process.

Subsequent studies expanded on these foundations by addressing the challenges posed by unsignalized, multi-lane, and occluded intersections. One approach integrated a risk-aware reward

Framework	Approach	Objective	Metrics
[Au and Stone, 2010]	Motion Planning with FCFS	Integrate motion planning to compute optimal arrival times and speeds under a basic FCFS rule	Average delay
[Liang et al., 2018]	FCFS Enhancement via Vehicle Clustering	Enhance FCFS by clustering vehicles to exchange information and adjust travel speeds dynamically	Average travel time, speed
[Liang et al., 2018]	Rule-Based Platoon Splitting	Split long vehicle platoons into sub-platoons to improve coordination and flexibility	Throughput, average delay
[Khayatian et al., 2019]	FCFS with Conflict Point Analysis and RTD Management	Account for wireless transmission delays using a PID controller to set actuation times, enhancing safety	Average travel time, waiting time
[Tachet et al., 2016]	BATCH Algorithm (Platoon-based Reordering)	Use a fixed reordering period to collect requests, thereby increasing capacity and reducing delays	Average delay, queue length
[Mitrovic et al., 2020]	Dynamic Lane Reconfiguration	Temporarily reassign lanes (e.g., opposite-direction use) to minimize conflicts and sustain uninterrupted flow	Throughput, lane utilization
[Xu, 2018]	1D Virtual Lane Scheduling via DFS	Project traffic flows onto a virtual 1D lane and synchronize vehicles using depth-first search	Crossing success rate, collision rate
[Fayazi and Vahidi, 2017]	MILP-based Centralized Control	Optimize crossing efficiency by solving a MILP to minimize arrival time variance	Delay, emission levels
[Feng et al., 2018]	Joint Signal and Trajectory Optimization	Integrate traffic signal control with trajectory planning to reduce emissions and delay	Emission levels, average delay
[Mirheli et al., 2019]	Distributed MINLP for Trajectory Planning	Enable vehicles to self-organize with a distributed MINLP that yields globally optimized trajectories	Travel time, throughput, average speed
[Guo et al., 2019]	Dynamic Programming for Joint Optimization	Jointly optimize vehicle trajectories and signal controllers to enhance safety and reduce fuel consumption and delay	Fuel consumption, average travel time
[Bisht and Shet, 2020]	V2X based platoon algorithm	Optimize platoon size based on the vehicle speed and information from intersection RSU	Speed, waiting time
[Kumaravel et al., 2021]	Job shop scheduling	Optimize platoon size and crossing sequence	Average travel time, average delay, fuel consumption
[Mushtaq et al., 2021]	Swarm intelligent	Reduce congestion and collision rate	collision rate
[Cong et al., 2022]	MILP	Optimize car-following order in platoons	speed
[Zhou et al., 2022]	constrained finite-time optimal control	Optimize virtual lane formation in a feedback loop	Average travel time, average delay, fuel consumption

Table 3.5: Summary of Heuristic-Based Optimization Methods for Isolated Intersection Management

function with a neural network-based Q-network to effectively estimate potential hazards and enable minimal yet impactful actions [Kamran et al., 2020]. Another method divided the complex task of intersection crossing into distinct sub-tasks such as straight driving and turning manoeuvres, utilizing a multi-task Deep Q-Network (DQN) to prioritize collision avoidance [Kai et al., 2020]. Similarly, [Li et al., 2022] introduced an adaptive autonomous intersection control model, named INTEL-PLT, which applies DQN to manage CAVs at signal-free intersections, integrating adaptive

platooning and reservation control. The method splits into two phases: first, a DQN learns platoon sizes by taking traffic density and intersection state as inputs, outputting discrete platoon configurations to regulate traffic flow; second, another DQN assigns reservation slots, using vehicle positions and speeds to prioritize crossings, aiming to minimize delays. Further advancements combined Monte Carlo Tree Search (MCTS) with a Double Deep Q-Network (DDQN) to improve the accuracy of action value approximations during planning, resulting in safer and more efficient navigation [Weingertner et al., 2020].

Building on these developments, later work introduced an attention-based driving policy that uses egocentric observations to focus on the most relevant spatial and temporal features. This selective attention mechanism not only optimized travel speed but also reduced collision risks and braking events [Seong et al., 2021]. Parallel efforts fused deep RL algorithms with model-based control strategies, such as merging Deep Deterministic Policy Gradient (DDPG) with Non-linear Model Predictive Control (NMPC), to better handle non-linear traffic constraints [Hamouda et al., 2021].

The most recent advancements have shifted towards collaborative decision-making, employing multi-agent frameworks that utilize Q-learning for adaptive, real-time responses to dynamic traffic conditions [Gunarathna et al., 2022, Huang et al., 2023]. Additionally, sequential optimization techniques integrated with RL have further refined intersection navigation, enhancing both safety and smoothness in vehicle manoeuvres [Tallapragada et al., 2023].

In summary, RL-based approaches for intersection management have demonstrated substantial potential in improving collision avoidance, driving efficiency, and adaptability to traffic dynamics. While most studies focus on simulated environments, the promising results suggest that RL could play a vital role in shaping the future of intelligent transportation systems, particularly in managing complex urban traffic networks.

3.2.4 Optimizing Crossing on Multiple Intersections

Optimising multiple intersections is a critical aspect of urban traffic management, requiring effective coordination across interconnected traffic networks. This section reviews heuristic-based and RL-based methods, which aim to enhance traffic efficiency, reduce delays, and ensure scalability in complex multi-intersection scenarios. Heuristic approaches leverage predefined strategies and optimization frameworks, while RL-based methods explore adaptive learning techniques to address dynamic traffic conditions however facing scalability challenges.

Heuristic-based Multi-Intersection Management

Efficient control of multiple intersections has evolved through a series of heuristic methods that gradually address the complexities of urban traffic coordination. Early foundational work introduced slot-based reservation protocols, and subsequent studies built upon that idea with increasingly sophisticated optimization and coordination techniques.

A pioneering AIM system [Dresner and Stone, 2004a] established a slot-based reservation protocol for coordinating autonomous vehicles without traditional traffic signals. This initial approach, originally developed for single intersections, was later extended by [Hausknecht et al., 2011] to

Framework	Approach	Objective	Metrics
[Demir and Sezer, 2018]	DRL with vehicle-specific dynamic and kinematic constraints	Incorporate inherent vehicle limitations into the RL decision-making process for robust intersection planning	Crossing success rate
[Kamran et al., 2020]	Risk-aware Q-network	Integrate a risk-aware reward function to estimate hazards and enable minimal, safe actions in occluded intersections	Collision rate
[Kai et al., 2020]	Multi-task Deep Q-Network (DQN)	Divide intersection crossing into sub-tasks (e.g., straight driving, turning) to prioritize collision avoidance	Collision rate
[Li et al., 2022]	Hierarchical task decomposition	Multi-policy DQN	Average travel time, fuel consumption
[Weingertner et al., 2020]	MCTS with Double DQN (DDQN)	Improve action value approximations during planning for safer and more efficient navigation	Crossing success rate
[Seong et al., 2021]	Attention-based Driving Policy	Utilize egocentric observations to focus on relevant spatio-temporal features for optimized speed and reduced braking events	Collision rate
[Hamouda et al., 2021]	Hybrid RL-NMPC Framework	Fuse deep RL with Non-linear MPC to better handle non-linear traffic constraints	Collision rate
[Tallapragada et al., 2023]	RL with Sequential Optimization	Enhance safety and smoothness in intersection navigation through sequential decision refinement	Arrival rate, throughput
[Gunarathna et al., 2022]	Multi-agent RL with Multi-discount Q-learning	Enable adaptive, real-time decision-making in dynamic traffic through decentralized coordination	Travel time
[Huang et al., 2023]	Collaborative Multi-agent RL	Balance short-term actions and long-term traffic objectives via cooperative agent strategies	Crossing time

Table 3.6: Summary of RL-Based Approaches for Intersection Management

manage a network of up to four interconnected intersections. Their simulations revealed that coordinating the reservation slots across adjacent intersections can significantly boost overall throughput and reduce travel delays.

Building on slot-based concepts, [Vasirani and Ossowski, 2011, Vasirani and Ossowski, 2014] reimagined urban traffic management as a distributed resource allocation problem. By allowing vehicle agents to bid for intersection capacity, their auction-based framework introduced competitive dynamics that balanced efficiency and fairness. Simulations on networks of up to 16 intersections demonstrated that dynamic bidding strategies could optimize resource allocation and improve traffic flow across larger urban networks.

A further evolution emerged with decentralized, multi-agent approaches. For examples, [Tlig et al., 2014] proposed a two-level system in which local intersection agents manage vehicle passage while a higher-level coordination layer synchronizes neighbouring intersections. Tested on networks of up to 10 intersections, this method showed notable improvements in traffic flow and reductions in waiting times, though scalability issues arose as computational complexity increased with network size.

[Zhang and Zhou, 2018] later combined Genetic Algorithms (GA) with dynamic programming to manage multiple adjacent intersections. Their strategy grouped vehicles into minimal units and generated compatible crossing patterns that were refined through GA optimization. Although simulations on networks of up to 12 intersections highlighted substantial traffic flow improvements, the computational burden posed challenges for real-time deployment under high-traffic conditions.

[Pei et al., 2021] extended earlier work by applying Monte Carlo Tree Search (MCTS) independently at each intersection to optimize real-time crossing orders. Evaluated on a network of four interconnected intersections, the MCTS method outperformed simple First-Come-First-Serve (FCFS) baselines. However, the absence of inter-intersection communication limited its ability to achieve global optimization, leaving room for further research on scaling MCTS to larger networks.

More recently, [Iwase et al., 2022] framed multi-intersection coordination as a Distributed Constraint Optimization Problem (DCOP). They designed a decentralized algorithm with polynomial-time complexity that, when applied to networks of up to 24 intersections, achieved a reduction in traffic delay of roughly 41% compared to managing intersections independently.

Despite these promising advances, several challenges persist:

- **Computational Complexity:** Approaches such as GA and dynamic programming demand extensive computational resources, which complicates their real-time use in high-density traffic. Additionally, methods that operate independently (e.g., some MCTS implementations) may struggle to capture global traffic dynamics.
- **Scalability:** As urban networks grow, decentralized systems and MCTS-based methods face increasing coordination demands and computational burdens. These issues limit the practical scalability of heuristic methods for extensive urban networks.
- **Real-Time Feasibility:** Even with polynomial-time algorithms like those used in DCOP, communication overhead and the need for rapid, adaptive computation remain significant obstacles to real-time deployment in highly dynamic traffic scenarios.

Framework	Approach	Objective	Scale	Metrics
[Hausknecht et al., 2011]	Extension of AIM to multi-intersection networks	Improve global throughput and reduce travel delays	4 intersections	Global throughput, travel delay
[Vasirani and Ossowski, 2011]	Auction-based distributed resource allocation	Enhance fairness and efficiency in resource usage	16 intersections	Travel time, bidding success rate
[Tlig et al., 2014]	Decentralized multi-agent system for CAVs	Minimize vehicle waiting times and enhance traffic flow continuity	5 intersections	Waiting time, traffic flow continuity
[Pei et al., 2021]	Distributed MCTS applied to multiple intersections	Optimize crossing orders in real-time	4 intersections	Crossing success rate, throughput
[Zhang and Zhou, 2018]	Genetic Algorithm (GA) with dynamic programming	Generate compatible crossing patterns for optimization	50 intersections	Computational efficiency, crossing success rate
[Iwase et al., 2022]	Distributed Constraint Optimization Problem (DCOP) framework	Reduce traffic delays and improve coordination efficiency	24 intersections	Average delay, coordination efficiency

Table 3.7: Summary of Heuristic-Based Optimization on Multi-Intersection Scenarios

RL-based Methods

We are not aware of any reinforcement learning (RL) method that achieves *fully unsignalized* control on multi-intersection networks larger than > 9 four-way intersections, other than our MCTS-based approach [Shi et al., 2024b]. Most unsignalized work stays at small scales—typically 2×2 or 3×3 grids (i.e., ≤ 9 intersections), such as a 2×2 multi-agent DQN [Zhang et al., 2020b]—or studies a single intersection with MARL variants [Cederle et al., 2024, He et al., 2024]. When researchers move to larger networks, they mostly study *signalized* adaptive traffic signal control (RL-ATSC) aimed at network-level coordination [Wei et al., 2019c, Zang et al., 2020a]. Even RL systems reported on 14+ intersections use *mixed* control instead of being fully unsignalized [Liu et al., 2025]. Recent surveys agree: RL for unsignalized intersections is still focused on single or small multi-intersection cases, and there is a lack of standard, reproducible benchmarks for network-scale testing [Zhong et al., 2020, Al-Sharman et al., 2024]. In short, for both scale and consistency of the control setting, a fully unsignalized RL solution for > 9 intersections is, to the best of our knowledge, still an open gap.

Summary

In conclusion, these studies (summarised in Table 3.8, Table 3.9 and Table 3.10) highlight the need for future research to focus on developing RL models that can generalise across a variety of intersection layouts and scale effectively to large networks.

Framework	Intersection Modelling			optimization Architecture			Optimization Strategy		Scale (No. of Intersections)
	Grid-based	Collision-based	Vehicle-based	Centralised	Decentralised	Hierarchical	RL-based	Heuristic	
[Bashiri et al., 2018]	✓			✓				✓	Single
[Dai et al., 2016]	✓			✓				✓	Single
[Wu et al., 2019b]	✓				✓		✓		Single
[Xu et al., 2020]	✓			✓				✓	Single
[Dresner and Stone, 2004a]	✓			✓				✓	Single
[Dresner and Stone, 2008]	✓			✓				✓	Single
[Belkhouche, 2019]	✓			✓				✓	Single
[Bian et al., 2019]	✓				✓			✓	Single
[Kamal et al., 2015]		✓		✓				✓	Single
[Lu and Kim, 2016]		✓		✓				✓	Single
[Katriniok et al., 2018]		✓			✓			✓	Single
[Katriniok et al., 2019]		✓			✓			✓	Single
[Li et al., 2018]			✓		✓			✓	Single
[Mirheli et al., 2019]			✓		✓			✓	Single
[Ge et al., 2022]			✓		✓			✓	Single
[Antonio and Maria-Dolores, 2022]			✓			✓	✓		Single
[Zheng et al., 2022]			✓		✓		✓		Single
[Gregoire et al., 2014]		✓		✓				✓	Single
[Lin et al., 2022]		✓			✓			✓	Single
[Katriniok et al., 2017]			✓		✓			✓	Single
[Yu et al., 2019]	✓			✓				✓	Single
[Mushtaq et al., 2021]			✓		✓			✓	Single
[Mitrovic et al., 2020]	✓			✓				✓	Single

Table 3.8: Summary of Unsignalized Intersection Management Frameworks (Part 1)

Framework	Intersection Modelling			optimization Architecture			Optimization Strategy		Scale (No. of Intersections)
	Grid-based	Collision-based	Vehicle-based	Centralised	Decentralised	Hierarchical	RL-based	Heuristic	
[Buzachis et al., 2020]			✓		✓			✓	Single
[Su et al., 2020]	✓			✓				✓	Single
[Li et al., 2021a]	✓			✓				✓	Single
[Wu et al., 2022]			✓	✓				✓	Single
[Ayeelyan et al., 2022]	✓			✓			✓		Single
[Abbas-Turki et al., 2023]		✓			✓			✓	Single
[Vu et al., 2018]		✓			✓			✓	Single
[Ali et al., 2020]			✓		✓			✓	Single
[Kai et al., 2020]			✓		✓		✓		Single
[Weingertner et al., 2020]			✓		✓		✓		Single
[Hamouda et al., 2021]			✓		✓		✓		Single
[Tallapragada et al., 2023]			✓		✓		✓		Single
[Morrissett et al., 2022]			✓		✓			✓	Single
[Li et al., 2023b]			✓		✓			✓	Single
[Wang et al., 2020a]			✓		✓		✓		Single
[Wu et al., 2019b]			✓		✓		✓		Single
[Antonio, 2022]			✓		✓		✓		Single
[Al-Sharman et al., 2023]			✓			✓	✓		Single
[Qiao et al., 2021]			✓			✓	✓		Single
[Kumaravel et al., 2021]	✓			✓			✓		Single
[Li et al., 2024]			✓			✓	✓		Single
[Li et al., 2023a]			✓			✓	✓		Single
[Zhou et al., 2022]	✓			✓				✓	Single
[Cong et al., 2022]		✓		✓				✓	Single

Table 3.9: Summary of Unsignalized Intersection Management Frameworks (Part 2)

Framework	Intersection Modelling			Optimization Architecture			Optimization Strategy		Scale (No. of Intersections)
	Grid-based	Collision-based	Vehicle-based	Centralized	Decentralized	Hierarchical	RL-based	Heuristic	
[Liang et al., 2018]	✓				✓			✓	Single
[Fayazi and Vahidi, 2017]	✓			✓				✓	Single
[Tachet et al., 2016]	✓			✓				✓	Single
[Xu, 2018]		✓		✓				✓	Single
[Mirheli et al., 2019]			✓		✓			✓	Single
[Guo et al., 2019]			✓		✓			✓	Single
[Feng et al., 2018]			✓		✓			✓	Single
[Kamran et al., 2020]			✓		✓		✓		Single
[Demir and Sezer, 2018]			✓		✓		✓		Single
[Seong et al., 2021]			✓		✓		✓		Single
[Gunarathna et al., 2022]	✓				✓		✓		Single
[Bisht and Shet, 2020]			✓		✓			✓	Single
[Li et al., 2022]	✓					✓	✓		Single
[Zhang et al., 2016]	✓				✓				2
[Huang et al., 2023]		✓		✓			✓		36
[Hausknecht et al., 2011]			✓		✓			✓	4
[Vasirani and Ossowski, 2011]			✓		✓			✓	16
[Tlig et al., 2014]			✓		✓			✓	5
[Pei et al., 2021]			✓		✓			✓	4
[Zhang and Zhou, 2018]			✓		✓			✓	50
[Iwase et al., 2022]			✓		✓			✓	24
[Xu et al., 2021b]	✓			✓			✓		4

Table 3.10: Summary of Unsignalized Intersection Management Frameworks (Part 3)

3.3 Research Gaps

Despite significant advances in intelligent traffic management, critical gaps remain that prevent the deployment of these systems in real-world urban environments. These gaps are particularly evident in unsignalized multi-intersection scenarios, where the complexity of coordination and scale presents substantial challenges.

3.3.1 Gap 1: Lack of Hierarchical Frameworks for Decentralized Traffic Management

Current reinforcement learning approaches in traffic management primarily focus on single intersections or small grid networks, failing to scale to city-wide traffic systems. The literature lacks comprehensive hierarchical frameworks that can effectively decompose complex traffic management problems into manageable components while maintaining system-wide coordination. Existing methods struggle to define appropriate state representations, action spaces, and reward functions across different hierarchical levels.

3.3.2 Gap 2: Absence of Standardized Intersection Representation Models

The literature lacks standardized methods for representing diverse intersection geometries and traffic conditions. Current approaches require manual configuration for each intersection type, making them difficult to deploy across heterogeneous urban networks. This limitation is particularly problematic for collision detection and trajectory planning, where the absence of unified geometric representations leads to computationally expensive, intersection-specific implementations.

3.3.3 Gap 3: Insufficient Integration of Individual and System-Wide Objectives

Current traffic management systems fail to effectively balance immediate vehicle-level objectives with long-term network performance goals. Existing reward functions do not adequately capture the relationship between individual vehicle efficiency and overall traffic flow, resulting in policies that optimize local metrics at the expense of system-wide performance. This problem is particularly severe in hierarchical systems, where the lack of proper credit assignment between layers leads to conflicting optimization objectives.

Current platooning and scheduling strategies exemplify this limitation. Existing platoon formation methods use overly restrictive grouping criteria [Tachet et al., 2016, Liang et al., 2018] and complex optimization procedures [Mushtaq et al., 2021, Bisht and Shet, 2020], leading to reduced intersection throughput and increased delays. These approaches fail to adapt to changing traffic patterns, resulting in rigid scheduling policies that perform poorly under varying demand conditions. The absence of adaptive scheduling mechanisms that can dynamically balance individual vehicle priorities with overall throughput represents a significant barrier to optimal intersection utilization.

3.3.4 Gap 4: Limited Training Methods for Large-Scale Multi-Agent Systems

The literature lacks effective training methodologies for hierarchical multi-agent reinforcement learning in large-scale traffic networks. Current approaches do not address the sample complexity

challenges of training deep hierarchical policies, where the joint state-action space becomes prohibitively large. While recent advances such as multi-agent proximal policy optimization (MAPPO) [Yu et al., 2022] and parallel neural Monte Carlo tree search (PNMCTS) [Shi et al., 2024a] show promise, their integration into unified hierarchical training frameworks remains underdeveloped.

3.3.5 Scheduling Under Specific Intersection Geometries and Traffic Flows

Prior studies on scheduling coordination at intersections have developed strategies that are often narrowly applicable to certain traffic conditions or intersection configurations, limiting their scalability and robustness. Heuristic-based methods may function adequately under stable traffic loads but perform poorly during peak hours or unforeseen disruptions. Rule-based approach reduces adaptability, which is critical for effective intersection management in complex urban environments [Li et al., 2019, Mahbub, 2020, Yao, 2020].

Other studies attempt to overcome this issue by employing RL-based policies; however, these are often computationally intensive and only feasible for managing a limited number of vehicle passing within specific intersection geometries [Wu, 2022, Antonio, 2022]. Such heterogeneity requires significant processing power to operate effectively, which limits their real-world scalability, especially in large, multi-intersection networks. This gap highlights the need for more efficient scheduling algorithms that balance computational efficiency with adaptability across a variety of intersection geometries.

3.4 Summary

This chapter reviewed the evolution of signalized traffic control systems, highlighting their advancements and limitations in addressing modern traffic challenges. The discussion then shifted to unsignalized traffic control systems, categorizing them based on modelling methodologies, optimization architectures, and management strategies. Heuristic-based methods, while computationally efficient, were found to struggle with scalability in heterogenous networks. SOTA RL-based approaches demonstrated adaptability but faced challenges with large-scale urban traffic network. The analysis underscored a critical need for scalable, adaptive, and computationally efficient frameworks to address the complexities of unsignalized multi-intersection traffic management.

4 Overall Design of the Hierarchical Multi-Agent Deep Reinforcement Learning Framework

Existing RL-based approaches for large-scale, unsignalized urban traffic network optimization are limited. This thesis aims to develop a learning-based solution that is not only efficient at individual unsignalized intersections but also capable of optimizing large-scale heterogeneous urban traffic networks.

Thus, the ultimate research question is:

Can a hierarchical, multi-agent reinforcement learning approach effectively optimize traffic flow in large-scale heterogeneous unsignalized urban traffic networks? Additionally, how does it compare to existing methods in terms of throughput, travel time, and travel-time reliability?

To answer these questions, the thesis addresses the following key research sub-questions:

1. **What is the appropriate hierarchical structure for a decentralized traffic management framework?**
2. **How can representation of intersection geometries be standardized?**
3. **How can the framework balance short-term individual goals with long-term traffic network objectives?**
4. **What are effective strategies for training hierarchical multi-agent systems for large-scale, heterogeneous urban networks?**

The following sections explain how each of these questions was addressed, discussing the motivation behind the proposed approaches, explaining why these approaches were expected to be effective and how, taken together, they address the overarching research question.

4.1 Motivation for and Structure of the Proposed HRL Framework

This section addresses the question: **What is the appropriate hierarchical structure for a decentralized traffic management framework?** First, the motivation for using a hierarchical

structure in large-scale urban traffic networks is discussed as well as the proposed structure. Then, each layer of the proposed decentralized HRL framework is introduced.

4.1.1 Motivation

Traffic coordination in large urban traffic networks spans multiple scales, from network-wide flow to individual vehicle paths, making a hierarchical approach essential. By decomposing the problem into layers, hierarchy facilitates structured decision-making, enabling specialized control at each level, while effectively managing interdependencies across the network. Learning methods are particularly well-suited for this context, as they enable agents to adapt to dynamic and unpredictable traffic patterns, overcoming the limitations of rigid, traditional heuristic-based approaches. For large-scale urban traffic management, hierarchical reinforcement learning (HRL) stands out as especially effective as discussed in Section 3.2.2: it integrates long-term strategic planning at the network level with short-term task accomplishment at intersection level, promoting scalable coordination among multiple agents in heterogeneous environments.

An important concept in this thesis is *platoon* [Jin et al., 2013, Bashiri and Fleming, 2017, Hu et al., 2018, Bisht and Shet, 2020, Mushtaq et al., 2021, Kumaravel et al., 2021], defined as a group of CAVs travelling in close proximity and crossing an intersection in the same direction. This concept is motivated by the need to enhance throughput and reduce congestion, allowing the platoon to be treated as a single, elongated vehicle for scheduling purposes. To manage short-term operations at the local level, the thesis introduces a centralized intersection scheduling agent. This agent optimizes crossing schedules for multiple arriving platoons, treating each platoon as a unified entity to ensure efficient and collision-free intersection management, detailed discussion is provided in Section 3.2.3.

Building on this, higher-level agents are motivated to optimize the broader traffic network. At the system optimization layer, decentralized agents deployed at intersections determine the number of CAVs in each platoon and, more critically, decide which platoons receive permission to cross. This decision-making process aims to maintain overall traffic efficiency. Drawing on established traffic pressure models [Wei et al., 2019a], traffic pressure at an intersection can be mathematically expressed as the difference between the sum of incoming vehicle densities and the sum of outgoing vehicle densities across all lanes. This metric captures congestion imbalances and provides a quantitative basis for intersection control. Based on this model, system optimization agents utilize a finer-grained traffic pressure calculation and dynamically adjust platoon lengths and crossing permissions to maximize throughput across the network. Additionally, these agents cooperate with neighbouring intersections by exchanging traffic pressure information, allowing for coordinated optimization of network-wide traffic dynamics.

Figure 4.1 illustrates how the system optimization agent and intersection scheduling agent are layered and deployed in a real urban network to take over traffic control from traffic lights. For each heterogeneous intersection, a system optimization agent is deployed on top of an intersection scheduling agent, where traffic dynamics are passed from the top to the bottom. Raw intersection information, i.e., CAV arrivals and traffic pressure, are abstracted, refined, and decomposed through the layers, with each layer assigned a specific task from the upper layer and resolving it. The

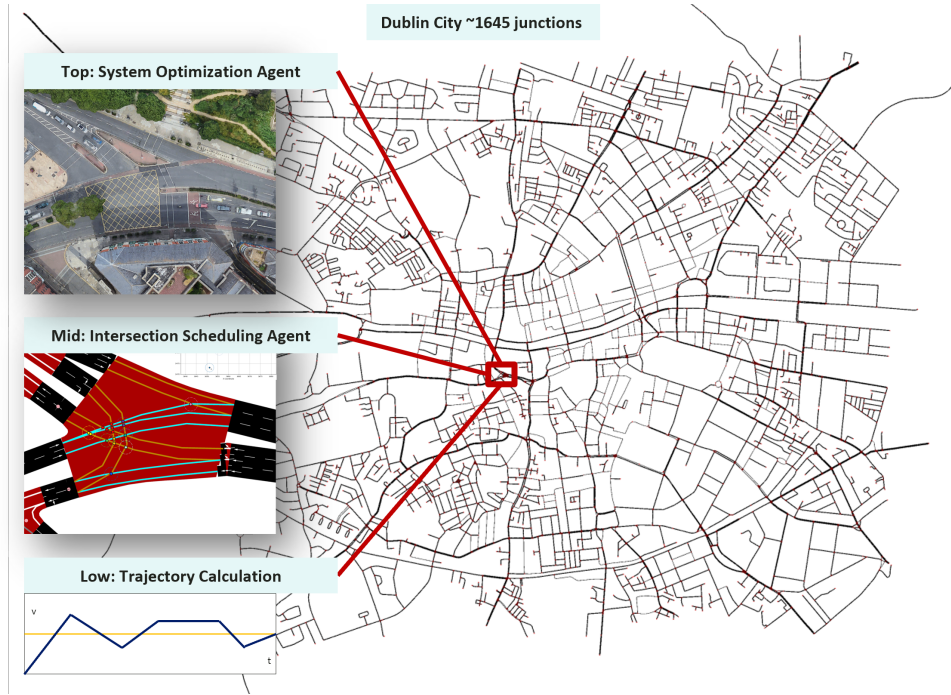


Figure 4.1: The system optimization agent and intersection scheduling agent are deployed in a layered hierarchy at intersections in the urban network.

information is progressively refined down to the third (last) layer, which consists of an analytical CAV trajectory calculation and execution model. The next section discusses the learning problem of each layer.

4.1.2 Proposed HRL Framework

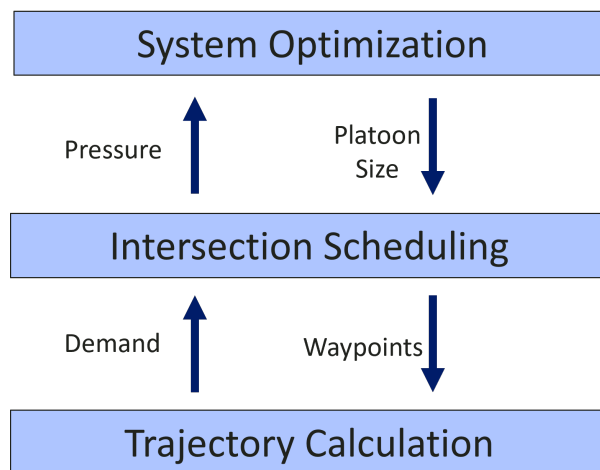


Figure 4.2: An abstract representation of each layer in the proposed architecture.

The architecture leverages a three-layer hierarchy to decompose the complex problem of traffic coordination into manageable sub-tasks, enabling scalable and adaptive control of CAVs. The abstract design of this three-layer architecture is presented in Figure 4.2 and detailed below:

1) **System Optimization Layer:** At the highest level, this learning-based layer deploys agents at

intersections to coordinate with neighbouring agents, collectively regulating platoon lengths and traffic throughput based on a finer-grained traffic pressure model. Vertically, traffic pressure originates from the intersection scheduling agent, influencing decisions once scheduling is completed. Horizontally, traffic pressure is also affected by demand from neighbouring intersections, ensuring a balanced traffic flow and improved throughput for system-wide benefits. The system optimization layer determines the platoon length for each lane and then passes this information to the intersection scheduling layer. A major objective for system optimization is achieving convergence to a mutually beneficial behaviour that optimizes travel-time reliability, reduces overall travel time, and enhances throughput across the network. The system optimization layer is comprehensively introduced in Chapter 6.

2) Intersection Scheduling Layer: At the mid-level, intersection scheduling agents operate under the system optimization layer. These learning-based agents are trained to be insensitive to variations in intersection geometry, enabling them to learn collision-free and near-optimal scheduling policies for heterogeneous intersections in a diverse traffic network. Once an intersection agent receives incoming platoon assignments determined by the system optimization layer, it generates near-optimal crossing schedules using proposed Parallel Neural Monte Carlo Tree Search (PNMCTS). The output is a list of waypoints then pass to the low-level trajectory calculation, that vehicles follow to navigate through the intersection efficiently. Requirements include high inference efficiency to produce schedules with low latency and adaptability to diverse intersection layouts. Challenges arise from achieving near-optimal scheduling for very busy intersection crossings, where high traffic density complicates collision avoidance. The intersection scheduling layer is thoroughly introduced in Chapter 5.

3) Trajectory Calculation Layer: At the lowest level, this layer analytically computes precise vehicle trajectories based on the provided waypoints, enabling CAVs to smoothly follow their assigned paths. Once trajectory execution is completed and new traffic arrives, the resulting change in traffic demand alters the traffic pressure at intersections. A key requirement is ensuring precise trajectory execution by CAV control systems to align with scheduled crossing times. The trajectory calculation layer is methodically introduced in Chapter 7.

4.2 Transforming Heterogeneous Intersections into a Uniform Representation

This section addresses the question: **How can the representation of intersection geometries be standardized?** First, an automatic transformation model is introduced that is used to convert heterogeneous intersection geometries into a uniform board-game-like representation. Then, the method for mathematically representing continuous traffic flow within this structured framework is discussed.

4.2.1 Automated Transformation of Heterogeneous Intersection Geometries

With platoons identified for crossing based on high-level decisions, an intersection representation is essential to efficiently model and allocate limited space. Heterogeneity plays a crucial role, as

intersections vary widely in geometry and traffic patterns, making scheduling inherently complex. This scheduling challenge can be formulated as a Job Shop Scheduling Problem (JSSP) [Ahn and Del Vecchio, 2016, Kaoud et al., 2017, Homayouni and Fontes, 2021], a well-known NP-hard optimization problem. In this formulation, each intersection is analogous to a job shop, where different platoon demands (jobs) must be scheduled across shared spatial and temporal space resources (machines) under constraints such as non-conflicting movements and safety margins. The complexity of this problem increases with the number of arriving vehicles and the scale of the intersection, requiring a uniform representation that can model diverse intersection layouts consistently while maintaining computational tractability.

The thesis proposes an intersection dynamics transformation model that automatically captures collision areas as snapshots using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering [Deng, 2020]. This method identifies potential conflict points based on intersection geometry and the coordination of intersecting points, grouping them into proximate clusters as collision areas. The detected collision areas are then used to convert heterogeneous intersections into a uniform board-game-like representation, where the intersection is discretized into manageable regions. This uniform model enables platoons' entry and exit timing requirements to be accurately described as they cross collision areas. Serving as the foundation for optimizing platoon movements, it allows platoons to dynamically adjust their entry timing, effectively preventing conflicts across intersections with varying geometries and layouts. The proposed board-game-like representation is introduced in Section 5.2, and the automatic collision-area detection method is presented in Section 6.1.1 and evaluated in Section 8.3.3.

4.2.2 Representing Continuous Traffic Flow

To effectively handle continuous traffic flow within this discretized representation, a Discrete-Time State Space Model (DTSSM) [Zhang et al., 2020a] is applied and comprehensively introduced in Section 5.1.2. The DTSSM formulates intersection control as a sequential decision process by defining state transitions at discrete time intervals, enabling a structured representation of vehicle movements. This discretization captures essential traffic dynamics, including platoon arrivals, departures, and platoon size as time progresses, while maintaining compatibility with the up-to-date scheduling status.

By integrating this model, the framework facilitates the exploration of optimal crossing sequences by leveraging state evolution over time. Additionally, the approach systematically incorporates ongoing platoon movements into the existing schedule, ensuring smooth coordination between successive time steps. This structured methodology accurately describes the evolution of intersection dynamics, making it well-suited for managing large-scale, unsignalized urban networks.

4.3 Balancing Short-Term and Long-Term Objectives Through Hierarchical Rewards

This section addresses the question: **How can the framework balance short-term individual goals with long-term traffic network objectives?** First, the learning objectives and challenges of

intersection scheduling are discussed. To further coordinate long-term platoon formation and throughput optimization, a hierarchical reward mechanism is designed to unify the objectives of both layers, ensuring seamless alignment of reward signals and facilitating joint optimization.

4.3.1 Schedule Continuous Vehicle Crossings at Unsignalized Intersections

To schedule uniform boards, this thesis proposes a centralized parallel neural Monte-Carlo tree search (PNMCTS) algorithm to determine platoon crossing priorities. The goal is to schedule platoon movements by learning to delay their entry into potential collision areas, avoiding conflicts with other platoons and ongoing previous schedule while minimizing total crossing time. A detailed example and explanation is provided in Section 5.1.

The first challenge lies in designing the action space. The primary objective of the action design is to ensure interpretability and maintain a reasonable number of actions that still captures a wide range of scheduling behaviours. Initially, an action space with dynamic values tied to the board state was considered. For instance, an action space of size 5 allows a platoon to be delayed by up to 5 steps, with the actual delay value adjusting dynamically based on the current state. This approach keeps the action space small, covering all possible states, but the trade-off is instability: the action values shift unpredictably as the state changes. This instability complicates the training process, causing the policy to struggle with convergence. To overcome this, we adopt a discretized action space consisting of 640 distinct actions. While this significantly increases the size of the action space, it provides consistent, interpretable actions that remain fixed throughout training. The downside is that training such a large action space requires careful design and fine-tuning to achieve optimal performance. Detailed action design and explanation is provided in Section 5.2.2.

The second challenge arises from the limitations of standard Neural Monte-Carlo Tree Search (NMCTS). While NMCTS is effective for many real-world scheduling tasks due to its extensive search capabilities and ability to evaluate potential crossing sequences [Mohamed et al., 2021], ensuring conflict-free and near-optimal schedules, it struggles with efficiency in this thesis's board setup. Intersection scheduling in the thesis configuration involves a vast number of potential initial states and demands rapid inference across continuous, high-volume traffic scenarios, which places significant strain on computational resources. To address this, thesis proposed PNMCTS that incorporates tailored four enhancements, which are entropy regulation, best-so-far resampling and curriculum learning techniques to boost schedule success rates, as well as short-path tree search to balance inference time and results quality. These modifications improve both the effectiveness and efficiency of the scheduling process. The proposed PNMCTS method is detailed in Section 5.3, while ablation studies analysing each modification is presented in Section 8.2.3. The performance evaluation of PNMCTS is discussed in the remaining parts of Section 8.2.

4.3.2 Hierarchical Rewards

To coordinate the short-term objective, i.e., intersection scheduling at the middle layer, and the long-term objective, i.e., platoon and throughput management at the system optimization layer, we propose a hierarchical reward model that combines long-term intersection traffic pressure reduction at high level with short-term solution quality feedback from the intersection scheduling agent. A

detailed discussion about the coordination reward design can be found in Section 6.2.5. This approach ensures that reward signals are informative at both hierarchical levels, improving the alignment of short-term and long-term objectives. Additionally, to foster cooperative decision-making in a multi-agent setting, each intersection agent incorporates the rewards of neighbouring intersections into its own reward function. By sharing reward information, agents can better coordinate their actions, preventing localized optimizations that may lead to network-wide congestion.

The hierarchical reward model mitigates sparse rewards by incorporating frequent feedback from the intersection scheduling agent, ensuring faster convergence and more stable learning. Furthermore, it enhances failure responsibility attribution by explicitly incorporating execution quality into the reward structure, allowing high-level policies to adjust based on real-time performance. The comparison of the proposed HRL framework with SOTA methods is presented in Section 8.3.1.

4.4 System-wide Optimization

This section addresses the question: **What are effective strategies for training hierarchical multi-agent systems for large-scale, heterogeneous urban networks?** First, the concept of platooning is introduced, and its role in traffic shaping and regulation is justified based on experimental results. Then, a decentralized multi-agent policy gradient method is described to optimize platoon grouping, leveraging neighbouring rewards and collective observations to enhance coordination and efficiency.

4.4.1 Platoon Method

The design of high-level network traffic optimization draws inspiration from signalized traffic pressure-based methods in [Wei et al., 2019a], where pressure measures queue length differences between incoming and outgoing lanes. However, pressure does not work well in unsignalized intersections because the goal shifts from managing queues to maintaining smooth traffic flow as vehicles approach, rendering queue-based metrics less effective. To address this difference, platoon length is determined by transforming traditional queue-length-based pressure into an adaptive mechanism that dynamically groups vehicles based on their proximity and real-time traffic conditions. However, an explicit assumption is that CAVs within a platoon share identical power outputs—such as acceleration, deceleration, and speed.

Experiments on the impact of platoons under different grouping policies were conducted in a 3×3 4-way 3-lane homogenous grid network with traffic volume of 1000/vehicle/h/road, with performance metrics for total throughput (TTP) and average travel time (ATT).

Results indicate that larger platoons do not always monotonically lead to lower ATT and higher TTP. However, to some extent, larger platoons enable the methods to regulate traffic flows better, providing learning-based approaches with flexibility to further reduce ATT while maintaining high TTP. A detailed analysis can be found in Section 8.3.4.

4.4.2 Multi-Agent Proximal Policy Optimization

To align platoon grouping with system-wide goals like travel time reliability and throughput, a decentralized MAPPO approach is adopted. Agents, each representing an intersection, coordinate by sharing platoon-crossing decisions with connected neighbours, using a **neighbouring reward model**, which consider the reward of interconnected intersections to optimize traffic flow collaboratively across areas (See Section 6.2.5). The proposed framework introduces a real-time sub-lane pressure model, which provides a finer-grained extension of the conventional lane pressure concept. This design further addresses heterogeneous intersection geometries, where a single incoming lane may diverge into multiple outgoing lanes, enabling a more accurate representation of traffic pressure (see Section 6.2.2) and promoting cooperative decision-making. This works because MAPPO's centralized training with decentralized execution allows agents to act independently without a central controller, enhancing scalability. The choice of PPO [Schulman et al., 2017a] is motivated by its robust performance and fast convergence in large-scale optimization problems [Xue et al., 2022, Pendyala et al., 2024]. However, isolated PPO is not enough in this thesis setup, as individual platoon decisions may conflict without network-wide coordination, reducing efficiency in heterogeneous networks. MAPPO may achieve improved coordination and overall traffic management performance, as demonstrated in complex multi-agent environments [Yu et al., 2022]. The efficiency of MAPPO and the neighbouring reward model is evaluated and tested in Section 8.3.4.

4.5 Low-Level Vehicle Trajectory Control with SUMO

The bottom layer focuses on precise vehicle trajectory control, employing a multi-segment acceleration control strategy to ensure that each CAV follows its assigned trajectory based on the crossing schedule provided by the middle layer. This fine-grained control is implemented in the simulation of urban mobility (SUMO) platform [Behrisch et al., 2011], which provides a detailed and realistic simulation environment for traffic scenarios. SUMO's capability to model individual vehicle behaviours and interactions allows for the implementation of precise control strategies. This integration helps bridge the gap between theoretical scheduling and practical implementation, enhancing the realism and applicability of the traffic management system.

4.6 Summary

The detailed hierarchical design of the proposed MARL framework (shown in Figure 4.3) effectively addresses the complexities of optimizing traffic flow in large-scale, heterogeneous, unsignalized urban networks with CAVs. Leveraging a multi-layered approach, the framework enhances throughput, reduces travel time, and improves travel-time reliability (See Chapter 8 for detailed evaluation). The top layer, i.e., system optimization agent, integrates MAPPO and traffic pressure methods for adaptive and scalable traffic optimization, generating platoon-based inputs for the lower layers (detail discussion in Chapter 5). The middle layer, i.e., the intersection scheduling layer, employs PNMCTS to efficiently schedule vehicle movements by transforming heterogeneous intersections into structured planning problems, enabling optimal trajectory generation (detail

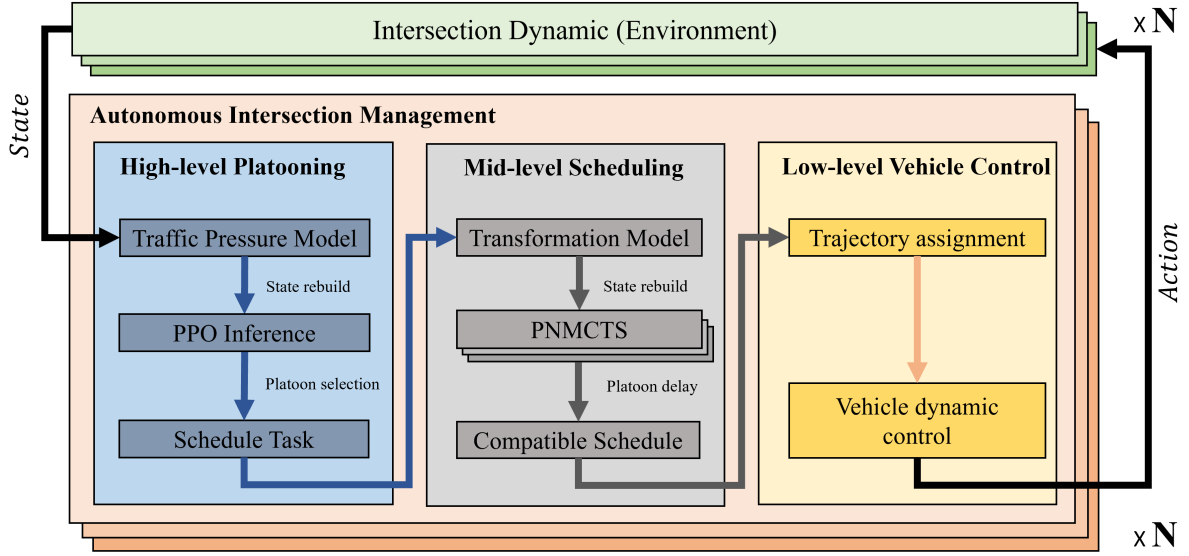


Figure 4.3: A detailed architecture design for the proposed decentralized hierarchical multi-agent deep reinforcement learning framework.

discussion in Chapter 6). Finally, the bottom layer, integrated with SUMO, ensures precise vehicle control by dynamically executing the planned crossings (detail discussion in Chapter 7). This cohesive framework not only optimizes large-scale urban traffic networks but also demonstrates superior performance compared to existing methods.

As evaluated in Section 8.3.2 for homogeneous grid network and Section 8.3.3 for large-scale heterogeneous city network, the proposed hierarchical MARL framework is thoroughly tested on a large-scale Dublin urban traffic network comprising 1645 heterogeneous intersections. The results, including zero-shot evaluation and fine-tuning for heavily congested areas in Section 8.3.3, confirm the framework's effectiveness in improving overall traffic efficiency and adaptability in complex urban environments.

5 Optimizing Intersection Scheduling

Efficiently managing vehicle crossings at intersections is a critical prerequisite for optimizing unsignalized urban traffic systems, particularly as traffic density and complexity increase. This thesis first studies how to efficiently schedule intersections, establishing a solid foundation for further multi-agent system-wide optimization. Additionally, it builds a concrete feedback mechanism for reward assignment, supporting long-term training and improving overall traffic management performance. This chapter addresses the research question: **How can the framework balance short-term individual goals with long-term traffic network objectives?** It formulates intersection crossing management as a short-term optimization task by proposing a scheduling method for vehicle platoons crossing intersections, emphasizing collision avoidance and optimized crossing times. The proposed framework employs an innovative modelling technique and state-enhanced decision-making algorithms to address the dynamic and continuous nature of intersection traffic.

The chapter begins by presenting the **Intersection Transformation Model** in Section 5.1, which translates the crossing dynamics of platoons into a structured board-like representation. This transformation identifies conflict points at intersections to cluster collision areas, enabling platoon scheduling for safe and efficient intersection crossings. The transformation model is built on a Discrete-Time State Space Model (DTSSM) [Zhang et al., 2020a] that provides a mathematical foundation for representing and evolving intersection states under varying traffic conditions.

The **Learning Environment Design** in Section 5.2 introduces the observation, action, and reward model design used to train a scheduling policy. The state representation captures the detailed intersection dynamics, while the action space is designed to resolve conflicts by precisely delaying vehicles when required. The reward model encourages shorter crossing times and efficient scheduling while penalising infeasible or inefficient solutions.

The chapter also details the **Training Method** in Section 5.3, which proposes Parallel Neural Monte Carlo Tree Search (PNMCTS) for decision-making and policy optimization. A dual-branch neural network architecture ensures robust feature extraction and valid action selection, while the parallel framework enhances computational efficiency and exploration diversity. Additionally, curriculum training is employed to progressively adapt policies from simple scenarios (empty intersections) to more complex ones (busy intersections), ensuring a stable and effective learning process.

5.1 Intersection Transformation Model

This section introduces the concept of dynamic intersection transformations and demonstrates how they can be mapped to a state space model allowing a formal representation.

5.1.1 Example Transformation of Intersection Dynamics

To analyse the crossing dynamics of unsignalised intersections, we start by addressing the scheduling of a fixed set of platoons crossing the junction. The core innovation lies in the application of a collision model that maps a sequence of resource requests —corresponding to the crossing of vehicles through potential collision area within the intersection — onto a game board-like representation.

In this intersection scenario (Figure 5.1), each approaching platoon is associated with a predefined trajectory to cross the junction. Along this trajectory, there are specific collision areas (labelled A–H in Figure 5.1 (left)), which represent points where the paths of different platoons may intersect. Assuming that each platoon crosses the intersection at its maximum permissible speed, we can predict an initial board configuration. This configuration indicates the time intervals during which each platoon occupies each collision area along its path.

These dynamics are then translated into a board-like representation, as illustrated in Figure 5.1 (right). This formalisation provides a clear and structured method for representing and managing the interaction between platoons, to enable efficient scheduling and collision-free navigation at unsignalised intersections.

The initial board reflects a scheduling strategy in which all incoming platoons cross the intersection at their maximum allowed speed. However, given that platoon paths may not always be compatible, there is a significant probability of conflicts occurring at conflict areas, represented as overlapping blocks on the board, e.g., platoons 3 and 4 conflict at D in Figure 5.1 (right).

To address this challenge, we adopt a strategy that delays (multiple steps of delay if necessary) the entry of specific platoons into collision areas, as illustrated in Figure 5.2. It is crucial to note that any delays introduced in entering a collision area propagate backward, affecting the entry times of platoons into preceding collision areas. These adjustments can be represented as revised scheduling boards. By modifying the schedule in this manner, we ensure that no two platoons are simultaneously present in the same collision area, thereby eliminating the possibility of collisions. The general problem can be defined as finding mutually compatible sequences of blocks that allow all platoons to cross the junction in the shortest possible time, given an arbitrary set of platoons to be scheduled.

For intersections where a previous schedule is already in progress —referred to here as 'busy intersections' — the dynamics can be transformed using the same state space model. This transformation addresses the random arrival of new platoons at busy intersections at arbitrary time points. An example of this is provided in Figure 5.3. It is important to mention that the newly arriving platoons may not only conflict with each other but may also conflict with the existing schedule, specifically the residual platoons still within the intersection. The presence of these

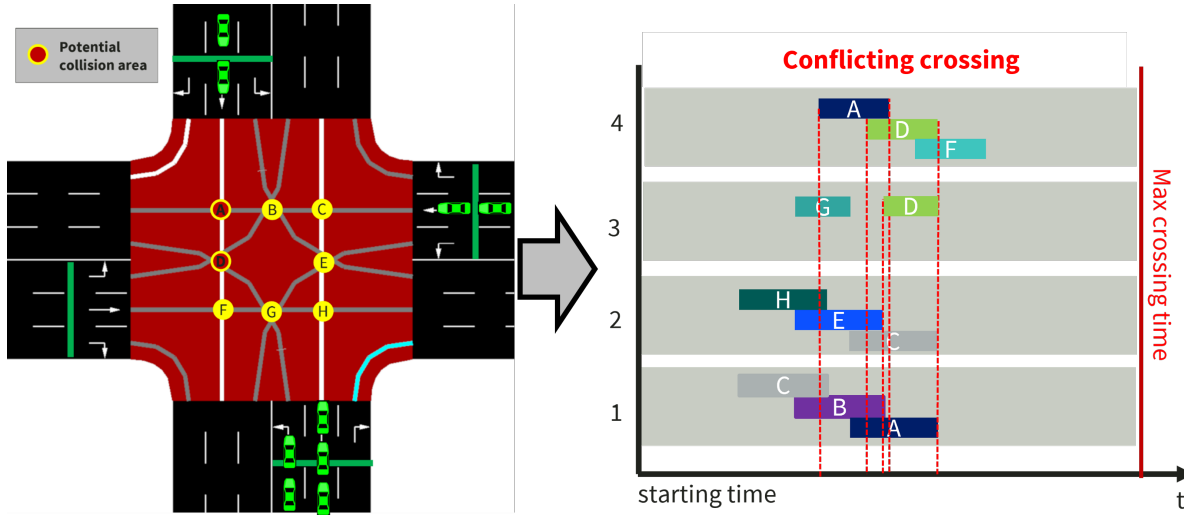


Figure 5.1: An intersection with four approaching platoons is illustrated on the left. The representative board corresponding to this scenario is given on the right. Entry and exit times for collision areas are shown on the time line (X-axis), while the Y-axis shows platoon numbers. As platoon 1 is moving forward, it will cross collision areas $C \rightarrow B \rightarrow A$ in that order, with the occupancy time of the corresponding collision area being mapped to time blocks (shown stacked for each platoon for readability). The occupancy time of collision areas C, B, and A by platoon 1 overlap because the platoon is longer than the gap between the collision areas. We can observe from the illustration that, for example, without a scheduling intervention, platoons 1 and 4 will collide in collision area A, Platoons 3 and 4 will collide in collision area D, as they would occupy it simultaneously.

residual platoons significantly increases the complexity of the scenario, potentially doubling the size of the solution space, as demonstrated in Figure 5.4.

5.1.2 Discrete-Time State Space Model

To further standardise the representation of the transformation model in a formal framework and facilitate its application to heterogeneous intersections (see Section 6.1), a DTSSM can be employed [Zhang et al., 2020a]. This approach enables the **recursive** representation of the state in a systematic and structured manner.

A state space model is a mathematical framework used to describe the behaviour of dynamic systems. The model consists of state variables and a state equation. The state equation describes how the state of the system evolves over time:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \quad (5.1)$$

where

- $\mathbf{x}_t$ is the state vector at time t .
- $\mathbf{A}$ is the state transition matrix.
- $\mathbf{u}_t$ is the control input vector at time t .
- $\mathbf{B}$ is the control input matrix.

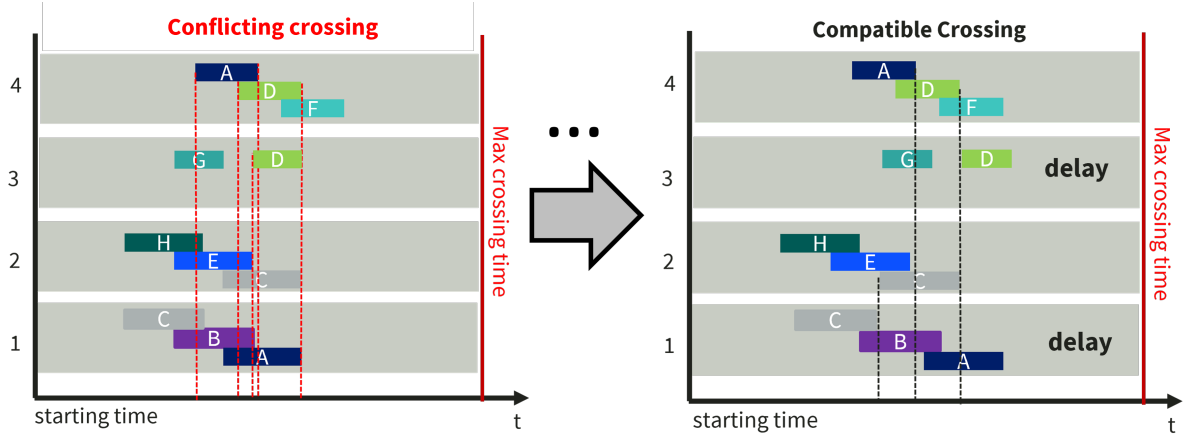


Figure 5.2: A potential solution to a conflicting board.

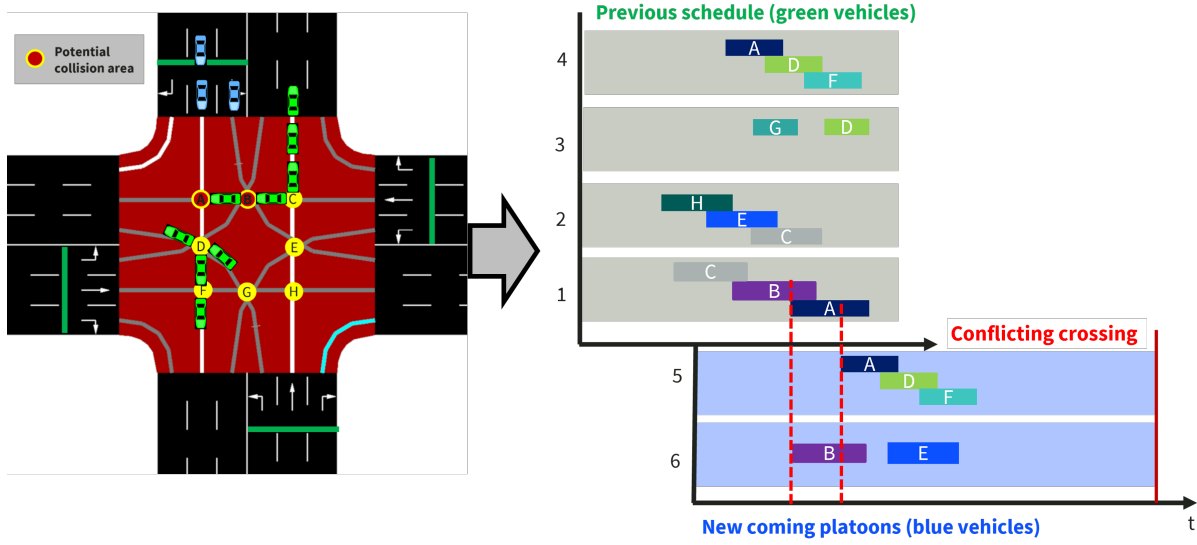


Figure 5.3: Transformation board represents a busy intersection that is executing a previous schedule (the upper board) and cannot be rescheduled. The new approaching platoons have potential to collide with the previous schedule at collision areas A and B.

- $\mathbf{w}_t$ is the process noise, representing uncertain dynamics.

In our design, the board evolution can be represented as follows,

$$\mathbf{S}_{t+1}^p = \mathbf{S}_t^p \oplus \mathbf{S}_t^{c*} \oplus \mathbf{w}_t \quad (5.2)$$

- $\mathbf{S}_t^p$ is the previously scheduled ('left-over') board at time t .
- $\mathbf{S}_t^c$ is the newly arriving board at time t .
- $\mathbf{S}_t^{c*}$ is the compatible schedule for $\mathbf{S}_t^c$ at time t , denoted as $\psi(\mathbf{S}_t^c | \mathbf{S}_t^p)$.
- $\oplus$ denotes a customised **overwrite** operation. Let A and B be matrices of the same dimensions and define the operation $\oplus$ such that the element-wise operation is defined as follows:

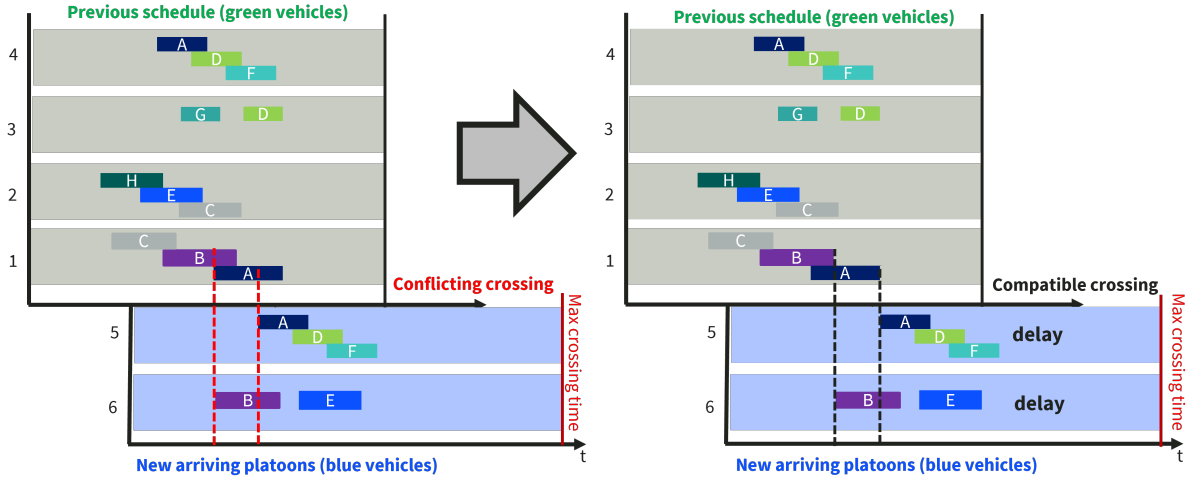


Figure 5.4: A potential solution to the conflicting board. The new schedule (for platoon 5 and 6) must ensure a compatible solution to the solved schedule (for platoon 1,2,3 and 4).

$$(A \oplus B)_{ij} = \begin{cases} B_{ij} & \text{if } B_{ij} > A_{ij} \\ A_{ij} & \text{otherwise} \end{cases} \quad (5.3)$$

where $(A \oplus B)_{ij}$ denotes the element at the i -th row and j -th column of the matrix resulting from the operation $A \oplus B$.

- $w_t = 0$ describes noise that could potentially sabotage the completion of the platoon crossing task, we assume that no uncertain dynamics are involved.

An elaborated evolution of the DTSSM is given in Figure 5.5.

- **Initial State and Concatenation:** Given a state S_0^P (previously scheduled board) and S_0^C (an empty board, i.e., no arriving platoons), the initial state S_0 can be formed by concatenating S_0^P and S_0^C , providing a starting point.
- **Merging Boards with $\oplus$ Operation:** If the two boards S_0^P and S_0^C are compatible (i.e., no time overlap on the same cell), they are merged using the $\oplus$ operation. The resulting unified board S_1^P maintains the same dimensions as the initial boards. The $\oplus$ operation is defined in Equation.5.3.
- **Time Update and Calibration:** After a time interval ΔT , the new platoons arrive, i.e., the board S_1^C are ready to be scheduled. The left-over board S_1^P needs to be calibrated by subtracting ΔT from each occupied cell. If any cell value becomes less than zero, it indicates that the corresponding collision area is unoccupied, then the cell is deactivated with default value of $(-\inf, -\inf)$. Mathematically, this can be represented as:

$$S_{t+\Delta T}^P = \max(S_t^P - \Delta T, 0)$$

- **Iterative Scheduling:** The updated board state S_1^P and the newly arrived board S_1^C are

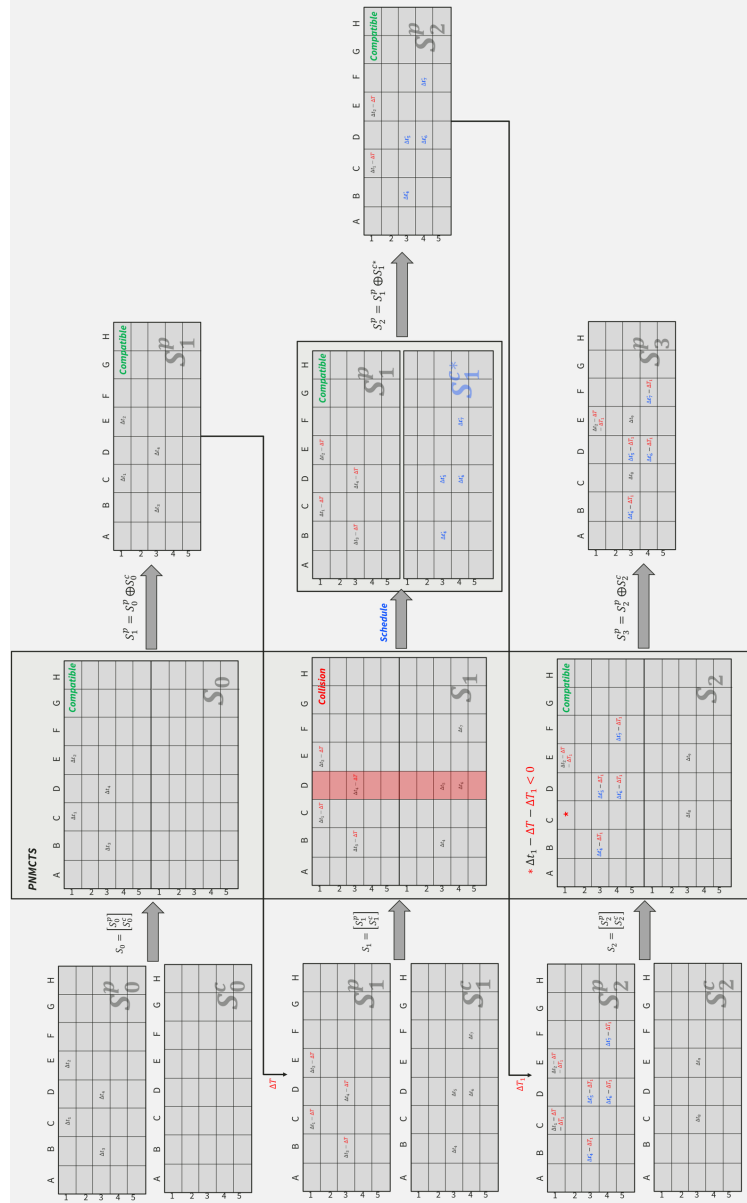


Figure 5.5: DTSSM design for intersection crossing.

concatenated and fed into the optimizer Ψ for rescheduling. The rescheduled board S_1^{c*} (solution) must remain compatible with S_1^p , as ensured by Ψ . The merging operation $S_1^{c*} \oplus S_1^p$ results in the next state S_2^p :

$$S_2^p = S_1^{c*} \oplus S_1^p$$

- **Capacity Threshold Analyse:** Multiple platoons are considered to be scheduled only when they arrive simultaneously, making it unlikely that the entire board capacity (row) will be consumed in one decision-making interval. Only vehicles travelling in the same directions are grouped into the same platoon, ensuring that the model accurately reflects traffic flow constraints and avoiding any unexpected behaviour. The capacity constraint is determined by the number of collision areas, which limits the column number, i.e., the size of board.

5.2 Learning Environment Design

The intersection scheduling problem, in general, involves identifying mutually compatible sequences of blocks that allow all platoons to cross the junction in the shortest possible time, given an arbitrary set of platoons to be scheduled. This section introduces the observation space, action space, and reward model used for learning optimised crossing schedules.

5.2.1 Observation Space

In the context of applying RL techniques for sequential decision-making, and leveraging proposed PNMCTS to discover optimal schedules, we formalise the definitions of the states, actions, and rewards required within the RL framework as follows:

	A	B	C	D	E	F	G	H
1	(18,23)	(15,20)	(12,17)					
2			(20,25)		(17,22)			(14,19)
3				(20,23)			(15,18)	
4	(16,21)			(19,23)		(22,25)		
5								

Figure 5.6: Grid representation of transformation board according to Figure 5.1.

The state is represented as a three-dimensional grid, as illustrated in Figure 5.6. The dimensions of the grid are determined by the maximum number of crossing platoons, P (e.g., 1–4), and the total number of collision areas, M (e.g., A–H). Each cell within the grid corresponds to a specific platoon and collision area, and contains a tuple of two values: $\langle t_{entry}, t_{exit} \rangle$. These values represent the entry and exit times of the P^{th} platoon with respect to the M^{th} collision area, capturing the platoon's occupancy duration in that collision area. If any occupancy durations overlap across different platoons, as shown in collision areas A and D in Figure 5.6, the board must then be resolved. This structured representation provides a detailed framework for analysing and scheduling platoon crossing sequences. The formal mathematical representation of the state is given in Equation (5.4).

$$S = \left\{ \left\{ \{ t_{entry}^{ij}, t_{exit}^{ij} \}_{i \in M} \right\}_{j \in P} \right\} \quad (5.4)$$

If a collision area is not part of the platoon's crossing path, the corresponding cell defaults to the placeholder value of $(-\infty, -\infty)$. This state representation comprehensively summarises all the essential geometric features and the dynamics of the intersection, including platoon size, travel speed, chosen path, distance to the intersection, and collision area distribution. Note that in the intersection geometry shown in Figure 5.1, a platoon can encounter at most three collision areas to

cross the intersection.

5.2.2 Action Space

To address overlaps on the board, it is sufficient to delay the entry times of (some) platoons into the collision areas that they intend to cross. Recognising that an optimal solution may involve imposing some delay on the entry or exit time of certain platoons into conflict areas, corresponding adjustments to their trajectories may be necessary. To facilitate this, we define discretised actions to delay a cell up to $n \in N$ moves. Each unit of delay, corresponding to one move, is set to 0.1 seconds. When an action is applied, the corresponding entry and exit times are delayed by $0.1 \cdot N$ seconds.

The action space for this problem is a set of discrete actions representing the possible delays that can be applied to each cell in the 8×8 grid. Formally, the action space can be expressed as:

$$\mathcal{A} = \left\{ \mathbf{a} \in \mathbb{R}^{8 \times 8 \times N} \mid a_{i,j,n} \in \{0.1 \times 1, 0.1 \times 2, \dots, 0.1 \times N\} \right\},$$

where:

- $i \in \{1, 2, \dots, 8\}$ denotes the platoon index,
- $j \in \{1, 2, \dots, 8\}$ denotes the collision area index,
- $n \in \{1, N\}$ is the number of delay steps, where each step corresponds to a delay increment of 0.1 seconds,
- $a_{i,j,n}$ represents a delay of $0.1 \times n$ seconds applied to the specific cell (i, j) ,

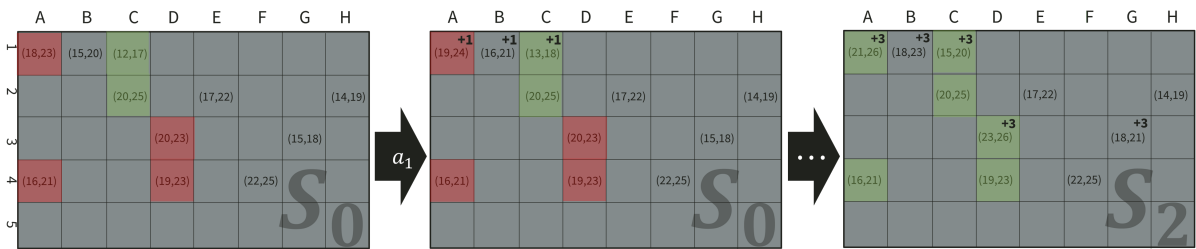


Figure 5.7: Transition of a board from its initial state to the solved state by applied a discrete action with 3 seconds delay.

The representation shown in Figure 5.7 illustrates the discrete actions applied to delay specific platoons in particular collision areas. For instance, the action $a_1 = (1, A, 10)$ delays platoon 1 in collision area A by $0.1 \times 10 = 1$ second, while the action $a_2 = (4, D, 5)$ delays platoon 4 in collision area D by $0.1 \times 5 = 0.5$ seconds. These delays are incrementally applied until the accumulated delay for platoon 1 in collision area A reaches 3 seconds, and the accumulated delay for platoon 4 in collision area D also reaches 3 seconds. At this point, the board is successfully solved.

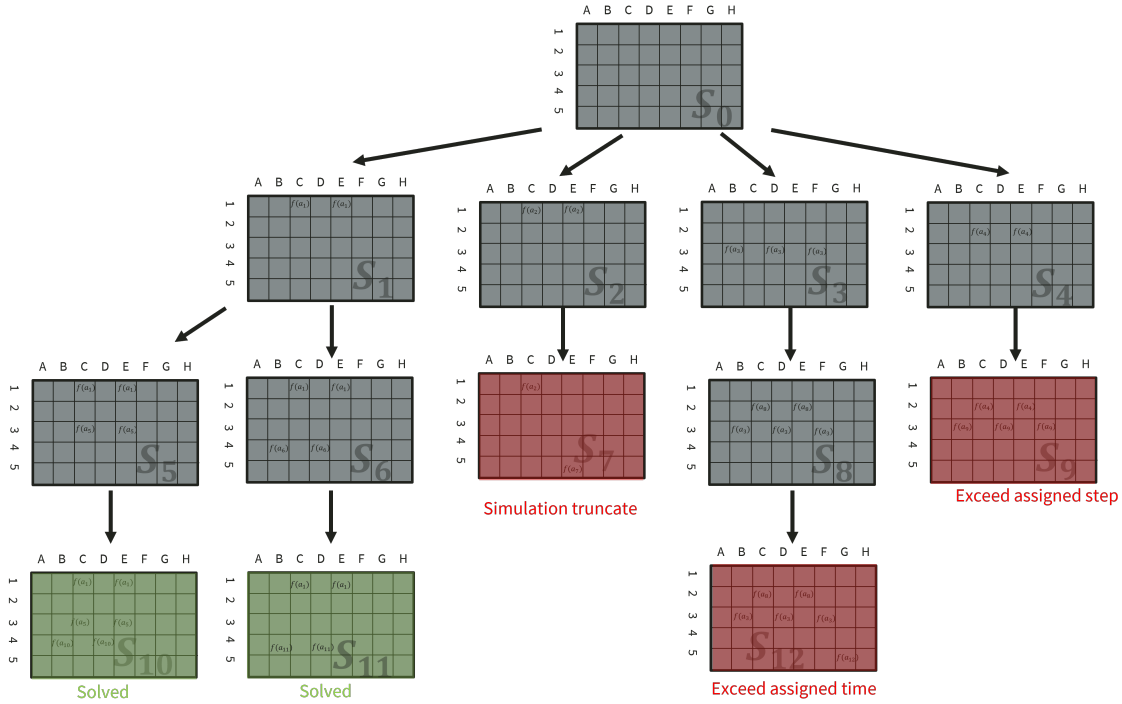
5.2.3 Reward Model

The reward model is formulated intuitively, as presented in Equation (5.5), to encourage the optimization of shorter crossing schedules:

$$r = \alpha \times \left(1 - \frac{t_{\text{cross}}}{T_{\text{max}}}\right) + (1 - \alpha) \times \left(1 - \frac{\text{step}}{S_{\text{max}}}\right), \quad (5.5)$$

where an upper limit is imposed on the crossing time, denoted as T_{max} . The reward decreases proportionally with the total crossing time t_{cross} for the final platoon, thereby motivating shorter crossing durations, $\left(1 - \frac{t_{\text{cross}}}{T_{\text{max}}}\right) \in (0, 1)$. Similarly, the second term considers the total number of steps step required to complete the crossing schedule. A higher number of steps results in a lower reward, encouraging solutions that minimise delays, $\left(1 - \frac{\text{step}}{S_{\text{max}}}\right) \in (0, 1)$.

The two components of the reward function are weighted by a parameter $\alpha \in (0, 1)$, allowing a trade-off between optimising for shorter crossing times and minimising the total number of steps. By construction, the reward value is constrained to lie within the range $(0, 1)$.



12

Figure 5.8: A demonstration of MCTS rollout from an initial board.

To prevent excessively long MCTS rollouts, additional constraints are applied, as shown in Figure 5.8: if the maximum crossing time is reached or the total number of delay steps exceeds a predefined threshold, S_{max} , the reward is penalised with a small negative value of -1×10^{-3} . Furthermore, if the crossing time exceeds the maximum permissible duration T_{max} (for a detailed study on determining the crossing cycle phase, refer to [Eom and Kim, 2020]), a significant penalty of -1 is assigned as the reward, effectively discouraging infeasible solutions.

For the purpose of training, the following parameter values are utilised: the maximum allowable crossing time is set to $T_{\text{max}} = 30\text{s}$, the step threshold is set to $S_{\text{max}} = 20$, the maximum delay step

per move $N = 10$, and the weighting factor α is set to 0.5. These values ensure a balanced reward function that effectively guides the search for optimal crossing schedules while maintaining manageable rollout length.

5.3 Training Method

This section introduces the proposed Parallel Neural Monte Carlo Tree Search (PNMCTS) algorithm, emphasizing its dual-branch neural network architecture tailored for efficient intersection scheduling tasks. The architecture integrates a primary feature extraction branch for policy and value prediction, and a secondary action masking branch to ensure valid and operationally feasible actions. Additionally, the section details a parallel experience collection strategy and various algorithmic enhancements, including gradient averaging, gradient clipping, entropy regularization, prioritized replay, and short-path MCTS rollouts. These enhancements collectively address challenges related to computational efficiency, training stability, and optimal decision-making performance under varying intersection traffic scenarios.

5.3.1 Parallel Neural Monte-Carlo Tree Search

PNMCTS Neural Network Structure

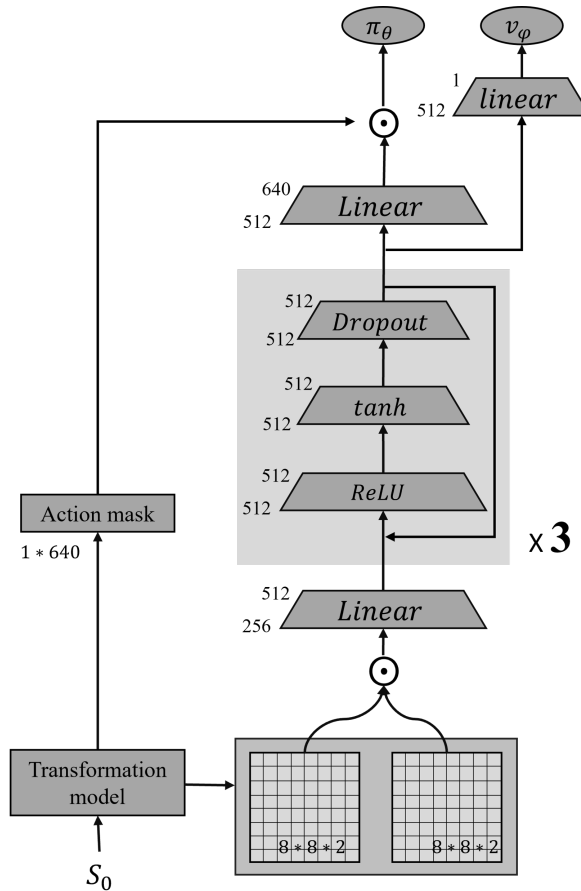


Figure 5.9: Dual-branch neural network architecture for board processing and action selection.

We employ a dual-branch neural network architecture designed to process board representations and

support action selection in scheduling tasks, as illustrated in Figure 5.9. The architecture comprises a feature extraction branch and an action masking branch, where the latter ensures valid decision-making by filtering out actions that exceed $T_{\max}$. The two input boards represent the previous schedule and the newly arrived demand, respectively.

In the main feature extraction branch, the transformed input state produced by the transformation model is first projected through a linear layer, followed by a stack of three fully connected layers with 512 hidden units each. These layers sequentially apply *ReLU* and *tanh* activations, with a dropout layer inserted afterward to improve generalization and stabilize training. Finally, the shared feature representation diverges into two output heads: one producing the policy π_θ and the other estimating the state value v_ϕ :

- **Policy Head:** The policy head outputs a probability distribution over all possible actions, which are used to guide the exploration process during the tree search.
- **Value Head:** The value head approximates the state value, providing a scalar estimate of the utility or quality of a given board state. This value is instrumental in guiding node selection within the MCTS framework, enabling the prioritisation of promising paths.

The secondary branch, referred to as the action masking branch, operates in parallel to the feature extraction branch. It directly processes the input board state to identify valid and invalid actions. This branch applies a clipping operation based on a predefined threshold, $T_{\max}$, which represents the maximum allowable delay for actions. The output of this branch is a binary mask, where:

- Valid actions, meaning those corresponding to the occupied cells in the grid, are assigned a value of 1.0.
- Invalid actions, specifically those associated with unoccupied cells, are assigned a value of $-\infty$.

This binary mask is subsequently applied to the output of the policy head. By incorporating the action mask, the softmax function ensures that valid actions are assigned positive probabilities, while invalid actions are set to zero. This mechanism ensures that the network's predictions remain constrained within the feasible action space, improving decision-making accuracy and efficiency.

Parallel Experience Collection

Unlike classic board games, the initial states in scheduling problems exhibit a high degree of variability. This is essentially a combinatorial optimization problem closely related to the dynamic job shop scheduling problem, which has been proven to be NP-hard, signifying that its computational complexities increase exponentially as the problem scales [Garey et al., 1976]. This variability poses a significant challenge, as training samples can differ substantially, making it difficult for the neural network to effectively 'remember' and stabilise the policy. Furthermore, another challenge arises during the collection of training examples: the tree search process can be computationally intensive, often requiring considerable time to identify a solution for a particular trajectory due to the vast number of possible outcomes at each step.

To balance performance and computational efficiency, constraints such as limited search depth and

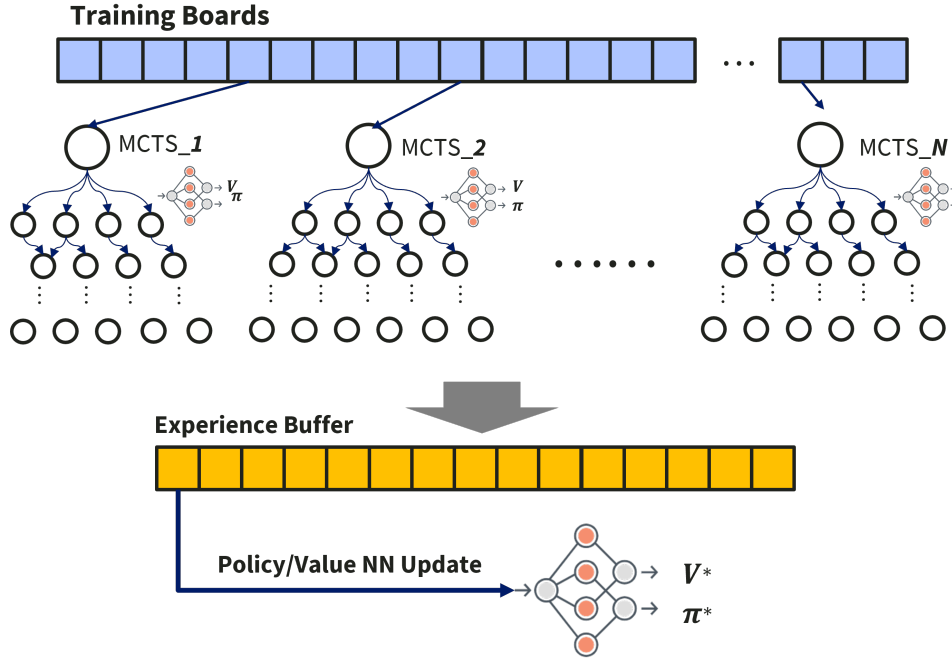


Figure 5.10: Training with parallel neural Monte-Carlo tree search.

maximum search durations are commonly applied to the tree search process [Świechowski et al., 2023]. However, imposing these limitations often prevents the Monte Carlo Tree Search (MCTS) from thoroughly exploring the search space, resulting in the failure to identify optimal solutions, particularly along shorter paths [Cowling et al., 2012]. Consequently, training samples are more likely to produce failure signals, increasing the risk of the policy prematurely converging to suboptimal local optima [Ma et al., 2019, Zhang and Lesser, 2020].

This thesis proposes a parallel multi-process framework for training derived from the standard Neural Monte Carlo Tree Search (NMCTS). This paradigm, referred to as PNMCTS, is depicted in Figure 5.10. In this approach, multiple processes are executed concurrently, with each process independently conducting an NMCTS. Each NMCTS instance is initialised with a random state or board S , and utilises a copy of the current policy network π and state value estimator V .

During the training process, each independent NMCTS progresses iteratively until the completion of the gameplay or episode. At the conclusion of all processes, the results from each NMCTS run are aggregated to update the network's policy and value estimates, resulting in an updated policy π^* and value function V^* . This parallelisation introduces several key benefits to the training process:

- **Enhanced Diversity:** The use of random initial states across processes ensures a diverse range of board configurations and state transitions. This diversity helps the network generalise better across varied scenarios.
- **Improved Exploration:** By running multiple independent NMCTS instances, the exploration of the solution space is significantly broadened, reducing the likelihood of converging to local optima.
- **Efficient Resource Utilisation:** The parallel execution leverages multi-core or distributed

computing resources, dramatically reducing the wall-clock time required for training.

- **Stabilised Learning:** The aggregation of diverse samples mitigates overfitting to specific trajectories, thereby contributing to the stabilisation of the policy network and value estimator.

PNMCTS Training Enhancements

Training the standard NMCTS on the proposed intersection board yields poor performance in terms of both success rate and crossing-time reduction in empty and busy intersection scenarios.

An *empty intersection* denotes a low-traffic scenario in which no residual platoons—i.e., previously scheduled groups of vehicles—are present. In such settings, the learning model primarily focuses on developing efficient and stable decision-making policies. However, tree search algorithms often generate high-variance trajectories, meaning that identical initial states can lead to solutions with widely varying delays composed of long or short steps. This variability can destabilize the learning process.

Conversely, *busy intersections* refer to scenarios where new platoons arrive at intersections that are already influenced by previously scheduled traffic. This phase introduces substantial complexity due to the presence of residual platoons, which significantly enlarge the action space and hinder the search for optimal trajectories, often resulting in a high rate of failed scheduling outcomes.

To address these challenges, this thesis proposes the following four enhancements:

- **Entropy regulation:** To mitigate the risk of gradient interference, gradients are accumulated over several mini-batches and averaged before updating the policy network. This is expressed in Equation 5.6, which uses cross-entropy loss to align the predicted policy distribution with the MCTS-derived target distribution.

The optimization objective $J(\theta)$ is given by:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N H(p_i, q_i) = -\frac{1}{N} \sum_{i=1}^N \sum_x p_i(x) \log(q_i(x)) \quad (5.6)$$

Here, $H(p_i, q_i)$ denotes the cross-entropy between the target distribution p_i (from MCTS) and the policy network's predicted distribution q_i for the i -th mini-batch, and N is the number of mini-batches over which the gradients are averaged.

Averaged gradients are then clipped to limit their magnitude, ensuring stability during the policy update. The clipping operation is formalised in Equation 5.7.

$$J(\theta) = \min(\text{clip}(\|\nabla J(\theta)\|_2, 1 - \epsilon, 1 + \epsilon)), \quad (5.7)$$

where ϵ is a temperature factor controlling the clipping range. These measures ensure that the model develops stable policies under low-variance conditions.

Finally, to encourage policy exploration and prevent premature convergence, an entropy regularisation term is added to the cross-entropy when calculating gradients, see Equation 5.8. This modified objective function $J'(\theta)$ is defined by Equation 5.9:

$$H(\pi(a|s; \theta)) = - \sum_i \pi_i(a|s; \theta) \log(\pi_i(a|s; \theta)) \quad (5.8)$$

$$J'(\theta) = J(\theta) - \beta H(\pi(a|s; \theta)), \quad (5.9)$$

where $H(\pi(a|s; \theta))$ denotes the entropy of the policy distribution π , and β controls the weight of the entropy term. By maximising entropy, the policy is encouraged to remain stochastic, aiding exploration in high-dimensional action spaces.

- **Priority replay mechanism:** To enhance the efficiency of the learning process, a prioritized replay mechanism is employed. This mechanism maintains a dedicated *best-so-far* archive, which stores the most effective trajectories identified during the search process for each specific board configuration. When a search attempt fails to discover a viable trajectory or encounters failure signals, the mechanism triggers the reintroduction of these *best-so-far* trajectories into the memory pool. This reintroduction occurs with a probability v , which balances the need to revisit proven solutions with the importance of exploring new alternatives. By revisiting previously effective trajectories, the model reinforces its understanding of successful strategies while simultaneously maintaining the capacity to explore novel and potentially superior solutions. This approach reduces the risk of forgetting critical information from past successes, ensuring it is retained throughout the iterative update process.
- **Curriculum learning:** Curriculum learning is a machine learning paradigm inspired by the structured way humans acquire knowledge. By progressively introducing tasks in an incremental sequence, starting with simpler ones and gradually increasing complexity, this approach facilitates more effective learning. In RL, curriculum learning is widely used to enhance sample efficiency and accelerate convergence, particularly in environments with sparse rewards and large exploration spaces [Yang et al., 2021]. By first training agents on basic tasks and then exposing them to progressively harder scenarios, agents develop foundational skills that transfer to more complex challenges [Narvekar et al., 2020].

The implementation of curriculum learning consists of several key components:

- **Task Sequencing:** Determining the order in which tasks are introduced ensures a smooth learning trajectory that aligns with the agent's evolving capabilities.
- **Difficulty Assessment:** Evaluating task complexity is essential to prevent agents from encountering tasks that are either too trivial or too challenging at a given stage.
- **Progressive Transfer Learning:** Knowledge gained from simpler tasks enables agents to better adapt to and solve increasingly complex problems, improving overall learning efficiency.

In the context of training intersection scheduling, curriculum learning is applied by gradually introducing more complex traffic scenarios into the training pool. Instead of abruptly switching between different difficulty levels, the training process begins with simple, unoccupied intersections, i.e., empty board and progressively incorporates more occupied intersections. As training progresses, more challenging intersection configurations—referred to as **busy boards**—are incrementally added to the training set. This gradual increase in complexity steers the policy toward solving more conflicts in the boards, allowing the model to adapt smoothly without abrupt performance degradation. Impact of such curriculum training strategy is provided in Section 8.2.3.

- **Short-path MCTS rollout:** Deep tree searches can be too slow for real-time inference, particularly when intersections are busy, straining computational resources and delaying schedule delivery, while pure network policy-based decision making could be insufficient and oversight. To address this, our framework employs short-path rollouts during NMCTS inference, adapting the search depth to the intersection's state—busy or empty—as shown in Figure 5.12. This method balances efficiency and accuracy, ensuring timely and reliable platoon scheduling.

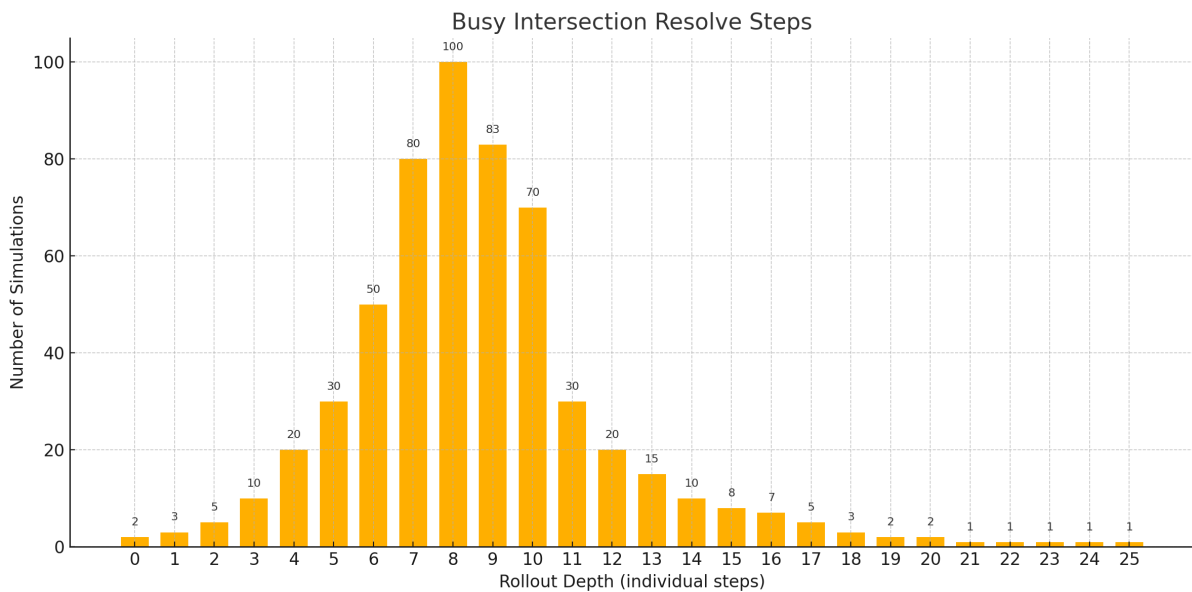


Figure 5.11: Distribution of steps to resolve busy boards.

The short-path rollout works by limiting the depth of NMCTS simulations based on the board's complexity. For busy boards, branch **b** triggers shallow searches, exploring only a few steps ahead, while branch **a** uses the trained network only for empty boards with fewer immediate conflicts. In busy scenarios, where many CAV platoons compete for limited intersection space, shallow rollouts prioritize quick decisions over exhaustive exploration, relying on the network's learned heuristics to maintain quality. The trained neural network guides these rollouts, predicting move values and policies to focus the search on promising sequences. The depth depends on traffic density and intersection complexity. For busy boards, a rollout depth of 5 to 10 steps is often sufficient, as the median number of solved busy board is 8, see Figure 5.11. This cooperation reduces the number of nodes evaluated,

cutting computational strain while still producing accurate schedules.

The method is inspired by AlphaZero [Silver et al., 2018], which uses limited-depth MCTS rollouts, typically stopping after a few moves, guided by a deep neural network that predicts outcomes and prioritizes branches. In Go, for example, it evaluates only about 80 moves per second—far fewer than the millions in the training phase—yet outperforms them by focusing on high-value paths. This efficiency and success in complex, real-time decision spaces justify adapting short-path rollouts for traffic scheduling.

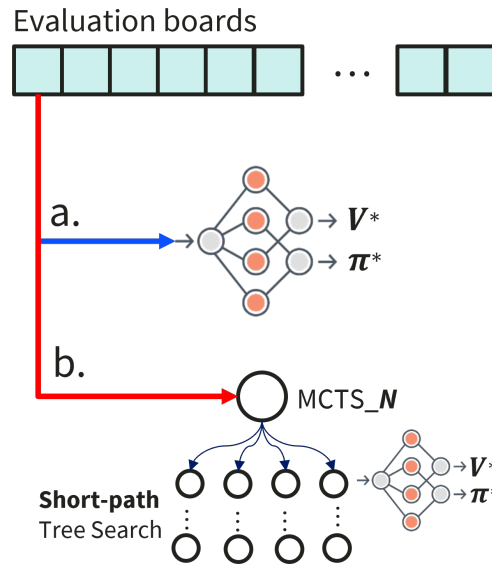


Figure 5.12: Short-path MCTS rollout: branch **a** for empty boards, branch **b** for busy boards.

Overall, the PNMCTS framework facilitates the collection of richer and more diverse training data, improving the robustness and performance of the neural network across a wide range of scheduling problems. By addressing the limitations of traditional NMCTS, this approach ensures more efficient and effective learning, enabling better solutions to complex decision-making tasks.

5.4 Summary

This chapter presented a comprehensive framework for scheduling vehicle platoons to cross unsignalized intersections, addressing the challenges of collision avoidance and traffic flow optimization. The proposed approach integrates a structured Intersection Transformation Model, which maps traffic dynamics into a board-like representation to visualise and resolve potential conflicts at collision areas.

To formalise the scheduling process, a DTSSM was introduced, enabling the systematic capture of the evolution of intersection states under varying traffic conditions. The state representation, action space, and reward model were carefully designed to encapsulate the complexities of intersection dynamics while encouraging efficient and safe scheduling. The action space allows for precise adjustments to platoon trajectories, and the reward model balances crossing time minimisation with system-wide efficiency.

This thesis proposes PNMCTS to train the uniform boards, which enhances exploration diversity and computational efficiency by running multiple tree searches in parallel. A dual-branch neural network architecture was utilised to extract features and ensure valid action selection, while curriculum training progressively adapted policies from simple to complex scenarios, ensuring stable learning outcomes. To further enhance performance on both empty and busy intersection scenarios, several improvements are applied to the PNMCTS algorithm. An ablation study evaluating the effectiveness of each enhancement is presented in Section 8.2.3.

By combining these elements, this design provides a scalable and adaptive solution to the challenges of managing unsignalized intersections. The integration of these innovative modelling techniques, robust decision-making algorithms, and efficient training methods ensures the framework's effectiveness in addressing realistic traffic patterns at scale.

Building upon the proposed PNMCTS, the integration of collision area detection with geometry-agnostic learning offers a comprehensive approach to optimizing traffic flow across diverse and complex networks. This solution is both practical and scalable, rendering it well-suited for deployment in large-scale urban environments.

6 Network-wide Traffic Optimization

While the previous chapter introduced a structured framework for scheduling vehicle platoons at individual intersections, real-world urban traffic systems require a more comprehensive, geometry-heterogeneity-resistant, network-wide approach. Optimizing a single intersection, though effective locally, does not account for variations in intersection geometry or dependencies between multiple intersections. As platoons move across the network, uncoordinated decisions at isolated intersections can create new congestion bottlenecks downstream, see Figure 8.20, reducing overall efficiency. To address these challenges, and answer the research question **What are effective strategies for training hierarchical multi-agent systems for large-scale, heterogeneous urban networks?** This thesis introduces a traffic optimization agent built on top of the intersection scheduling agents, incorporating two key advancements: 1) heterogeneous intersection recognition and training, and 2) multi-agent coordination and planning.

To enable optimization of heterogeneous intersections, this thesis proposes two innovations: 1) an automatic collision area detection mechanism for unsignalized intersections (Section 6.1.1) and 2) a geometry-agnostic learning (GAL) framework designed for scalable training to enhance inference generalization (Section 6.1.2).

For multi-intersection coordination, we introduce a decentralized Multi-Agent Proximal Policy Optimization (MAPPO) based approach, enabling system-wide traffic optimization through cooperative learning among intersections. MAPPO allows multiple intersections to dynamically regulate vehicle platoon formations and adapt to evolving traffic conditions in real time, then determined platoons are passed to lower intersection scheduling for compatible cross. We begin by describing the **Learning Environment Design** in Section 6.2, which expands on the intersection transformation concepts introduced in the previous chapter. This representation incorporates traffic pressure, virtual lanes, and lane-level state encoding to capture complex interactions between intersections using a sub-lane traffic pressure model. Embedding these dynamics in a decentralized framework ensures traffic optimization agents make informed, coordinated decisions that contribute to global traffic optimization. To align agent objectives with system-wide efficiency, we introduce the **Reward Model** in Section 6.2.5. Building on previous intersection scheduling principles, this model integrates feedback from the intersection scheduling agent (see Section 5.3), ensuring that improvements at one intersection do not negatively impact others. Next, we present the **Training Method** in Section 6.3, which applies MAPPO's approach to centralized training with decentralized execution (CTDE). This structure enables autonomous intersection operation while maintaining a shared learning framework for network-wide traffic coordination. With a shared global value network

and independent policy networks, this approach ensures both scalability and coordination, making it suitable for large-scale urban environments.

By introducing a multi-agent decentralized layer on top of the intersection scheduling layer, this chapter advances a more holistic traffic management strategy. The integration of the sub-lane traffic pressure model, cooperative reinforcement learning, and distributed decision-making offers a scalable, efficient, and adaptive solution for future urban traffic networks.

6.1 Training Scheduling Policy for Large Heterogeneous Networks

The optimization of heterogeneous networks presents significant challenges due to the diverse geometries, traffic dynamics, and coordination requirements inherent in large-scale urban networks. The thesis introduces methodologies designed to address these challenges, focusing on two key aspects: the automatic detection of collision areas in unsignalised intersections and the development of a GAL framework for scalable traffic management.

6.1.1 Automatic Detection of Collision Areas in Unsignalised Intersections

To identify intersecting points within a junction, a topology-based approach is applied using intersection geometry description files, e.g., SUMO network files (`.net.xml`). This step identifies all raw intersected points inside the junction as **collision points**, which are then refined through a clustering process that merges nearby points into cohesive **collision areas**, reducing complexity and improving computational efficiency.

To effectively identify and merge proximate collision points into collision areas, a robust clustering method is required to handle varying intersection geometries and traffic patterns. Among various clustering techniques, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Deng, 2020] was chosen due to its ability to detect arbitrarily shaped clusters without requiring a predefined number of clusters. This property makes DBSCAN particularly well-suited for traffic environments, where intersection layouts and vehicle trajectories exhibit significant variability.

A key advantage of DBSCAN is its adaptability to different intersection geometries through the tuning of the clustering radius ϵ . By adjusting ϵ , the granularity of collision area detection can be controlled. Larger ϵ merges more collision points into fewer collision areas, simplifying scheduling but potentially sacrificing precision. This results in more conservative safety margins at the cost of reduced crossing efficiency. Smaller ϵ increases the number of detected collision areas, allowing finer-grained conflict resolution but introducing additional computational overhead.

A radius of ϵ meters (the *epsilon* parameter in DBSCAN) is chosen as the threshold distance for clustering collision areas. This selection is guided by the critical *Time-to-Collision (TTC)* threshold, typically set at 1.5 seconds for collision avoidance [Das and Maurya, 2019], balancing detection accuracy and computational efficiency. It ensures meaningful grouping of collision points while maintaining manageable scheduling complexity. Detailed derivation of ϵ is provided in Section 8.3.3. Although dynamically optimizing ϵ for each junction based on its unique configuration remains an open challenge, it presents an avenue for future research in adaptive collision area detection.

The clustering process results in distinct collision areas, as demonstrated in Figure 8.26a, which depicts a junction in Dublin City with an 11 incoming virtual lane configuration. This specific junction yields nine unique collision areas post-clustering. Figure 8.27 provides a comprehensive view of all auto-detected collision areas in the broader Dublin city network.

6.1.2 Geometry-Agnostic Learning for Scalable Policy Generalization

In urban traffic networks, diverse intersection geometries pose a challenge for scalable policy learning. Most reinforcement learning (RL)-based methods rely on geometry-specific models or require retraining for each layout, limiting *generalization* and increasing computational costs, making large-scale deployment impractical.

To address this challenge, we leverage the proposed **Discrete-Time State Space Model (DTSSM)-based intersection representation** (introduced in Section 5.1.2), which provides a structured and uniform way to represent intersection geometries. In this formulation, intersections of varying layouts are mapped onto a *board-like representation*, where different geometric structures occupy different cells, forming unique but comparable spatial patterns. This abstraction allows us to model heterogeneous intersections within a *shared representational space*, eliminating the need for geometry-specific training.

Building upon this unified representation, we propose a **Geometry-Agnostic Learning (GAL) framework**, which enables a single policy to learn across unsignalized intersections with different layouts without explicit dependence on their specific geometry. By abstracting intersection structures through *DTSSM transformation*, GAL facilitates *transferable policy learning*, significantly enhancing adaptability across diverse urban environments.

A key advantage of GAL is its ability to support both zero-shot generalization and efficient fine-tuning. Zero-shot generalization allows policies trained on known intersections to be directly applied to unseen geometries without requiring additional retraining. By learning a policy that generalizes across different intersection structures, GAL reduces the reliance on extensive retraining and improves adaptability.

In addition to zero-shot generalization, GAL enables efficient fine-tuning when adapting policies to complex intersections with unique constraints. Instead of training from scratch, GAL leverages prior knowledge from learned representations, allowing the model to quickly adjust specific board patterns with minimal additional training data. This reduces computational overhead while maintaining high adaptability, making GAL a scalable solution for large-scale urban traffic networks. Detailed experimental results are provided in Section 8.3.3.

Let $\mathcal{T}$ represent the set of all junction geometries within a specific urban area, where each unique geometry $i \in \mathcal{T}$ corresponds to a distinct junction type characterized by its own state space $\mathcal{S}_i$ and action space $\mathcal{A}_i$.

The unified state space $\mathcal{S}$ and unified action space $\mathcal{A}$ are defined as the union of all individual state

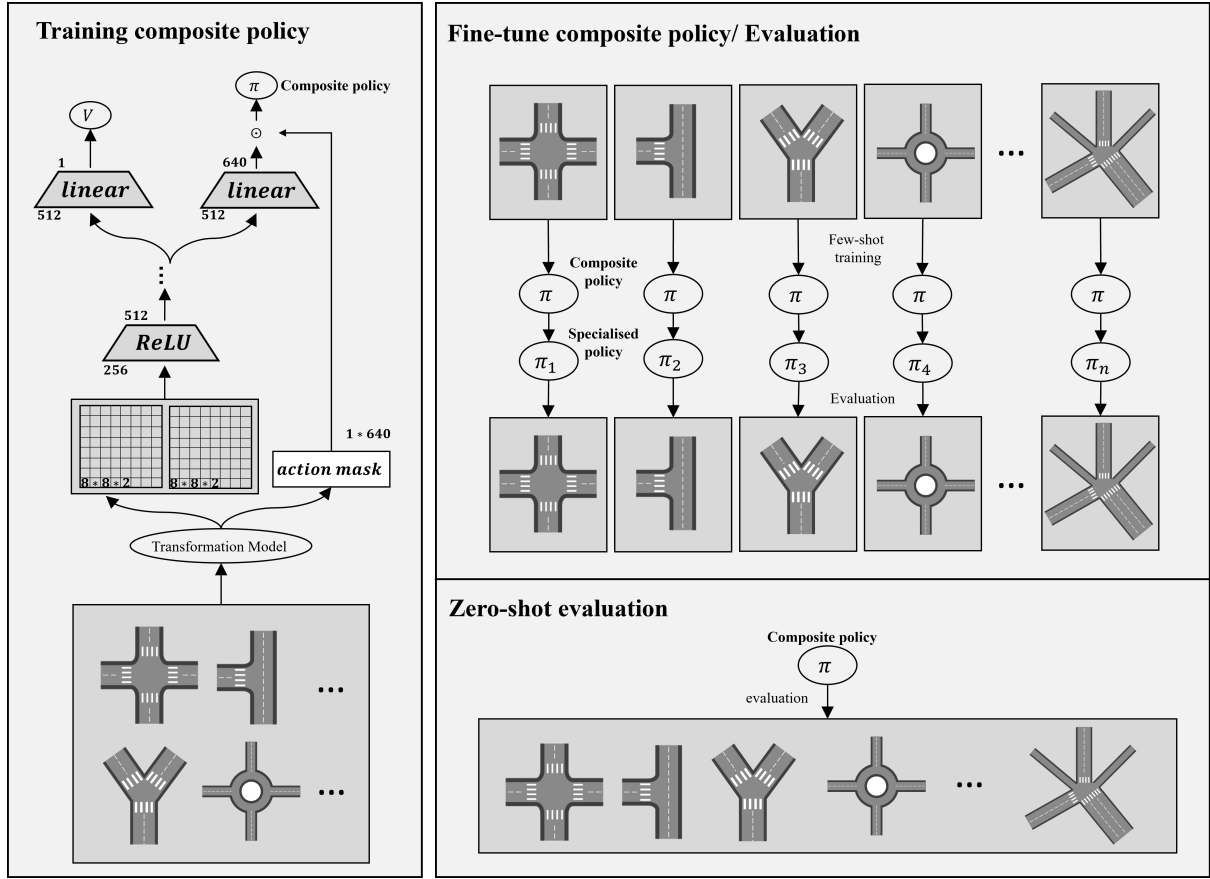


Figure 6.1: An urban geometry-agnostic learning approach. Composite-policy training process (left), evaluation with fine-tuned policy (top-right), and zero-shot evaluation (bottom-right).

and action spaces, respectively:

$$\mathcal{S} = \bigcup_{i \in \mathcal{T}} \mathcal{S}_i \quad \text{and} \quad \mathcal{A} = \bigcup_{i \in \mathcal{T}} \mathcal{A}_i$$

Figure 6.1 illustrates the training process for the GAL policy. The composite policy, denoted as π_θ , is trained on board representations sampled uniformly from the unified state space $\mathcal{S}$. This unified state space is constructed using the proposed DTSSM-based intersection transformation model, which standardizes diverse urban junction geometries and traffic dynamics into a common board-based format.

Building upon the previous chapter, where the proposed Parallel Neural Monte Carlo Tree Search (PNMCTS) algorithm was introduced for intersection scheduling, we now focus on enhancing its applicability to diverse and large-scale urban scenarios. Specifically, this section addresses the need for a unified representation of intersections with varying geometries to ensure consistent decision-making and policy generalization across the traffic network.

To balance the spatial coverage required by large intersections with the representation density needed for smaller ones, we adopt a uniform board size of 8×8 . This configuration successfully represents 1560 out of 1645 intersections—approximately 95%—in the test environment of Dublin city.

For any given intersection geometry $i \in \mathcal{T}$, the corresponding state space $\mathcal{S}_i$ and action space $\mathcal{A}_i$ are mapped into a unified state space $\mathcal{S}$ and action space $\mathcal{A}$. The state mapping is based on the occupancy of collision areas, which are activated by platoon movements specific to each intersection's layout. Similarly, the action space is standardized by defining actions as delay steps applied to these collision areas. These delay steps can be converted into real-world delay times (in seconds), enabling consistent policy application across heterogeneous intersections with varying numbers and configurations of incoming platoons.

When applying the proposed PNMCTS framework, the objective function for the intersection scheduling agent can be reformulated as:

$$\max_{\theta} \mathbb{E}_{i \sim \mathcal{T}} \left[\mathbb{E}_{(s,a) \sim (\mathcal{S}_i, \mathcal{A}_i)} [Q^*(s, a; \pi_{\theta})] \right],$$

where $Q^*(s, a; \pi_{\theta})$ denotes the expected cumulative reward of a state-action pair (s, a) , estimated through MCTS exploration guided by the composite policy π_{θ} derived from the GAL framework. By optimizing π_{θ} to maximize the expected reward over a distribution of intersection geometries, the framework enables generalizable and scalable intersection scheduling across the network.

A key advantage of the proposed GAL design is that it eliminates the need for reversing mapping—from the unified board back to the real-world. Since the same DTSSM is used to transform new heterogeneous intersection geometries into a uniform board representation, PNMCTS can perform inference directly within this standardized space, eliminating the need for geometry-specific control outputs. While this abstraction reduces the need for geometry-specific policies and promotes generalisation, it introduces several limitations. First, the transformation from specific geometries to the unified board representation may introduce sparsity in the encoded information. Specifically, to accommodate large intersections within a fixed 8×8 board structure, smaller intersections occupy only a limited portion of the board, resulting in sparse state representations. This sparsity may degrade the learning efficiency and decision accuracy of the policy. Second, although GAL enables trained policy transfers across unseen geometries, its performance may degrade on highly irregular layouts that differ significantly from the training distribution. Such cases highlight the limitations of relying solely on a unified representation, as extreme geometric variations may introduce interaction patterns that are not well-captured during training. To address these challenges and ensure efficiency in highly heterogeneous urban environments, further enhancement of the intersection transformation model and the integration of adaptive fine-tuning mechanisms are introduced.

After training the composite policy on the geometry-agnostic board representation, two complementary evaluation strategies are adopted to balance generalisation and adaptability: *zero-shot evaluation*, which tests the policy's ability to directly generalise to unseen intersection geometries without additional training, and *fine-tuning*, which quickly adapts the composite policy to a new geometry.

Zero-Shot Evaluation

Zero-shot evaluation offers a direct evaluation of the generalisation capability of the composite policy without requiring any additional training. In this mode, the pre-trained composite policy is deployed on the network, relying entirely on the knowledge acquired during the geometry-agnostic training phase. The ability of the composite policy to generalise across various junction configurations and traffic scenarios is demonstrated in Section 8.3.3.

Fine-Tuning with Few-Shot Training

Few-shot fine-tuning is an effective method in RL for adapting a pre-trained policy to specific tasks or environments using minimal additional data. In the context of urban junctions, this involves adapting the composite policy π_θ to a particular junction geometry, denoted by $i \in \mathcal{T}$. As each specific geometry typically activates a distinct subset of cells within the board representation during the training process, certain activation patterns associated with the geometry might not be sufficiently emphasised in the generalised composite policy. Fine-tuning provides an opportunity to adjust the policy parameters to align more closely with these geometry-specific activation patterns. By reactivating these patterns more consistently during the adaptation phase, the policy can better capture the nuances of the specific geometry, ultimately resulting in improved performance for tasks associated with that geometry.

The fine-tuning process utilises interaction data, represented as tuples $\langle s_{i,t}, a_i, s_{i,t+1}, r_{i,t+1} \rangle$, where $s_{i,t}$ is the state at time t , a_i is the action, $s_{i,t+1}$ is the subsequent state, and $r_{i,t+1}$ is the received reward. These trajectories are gathered by the simulation of the target intersection, which provide insights into the unique activation patterns on the board caused by the specific geometry of the junction. Through a limited number of training iterations, the policy adjusts its parameters to capture the fine-grained behaviours required for the target geometry while leveraging the general knowledge encoded in the pre-trained composite policy.

6.2 Learning Environment Design for Network-wide Traffic Optimization

In this section, we introduce the concept of fine-grained virtual lanes (sub-lanes) along with a detailed traffic pressure model to achieve a more precise representation of intersection congestion, thereby capturing traffic patterns more effectively. To integrate decision-making for both intersection scheduling and network-wide traffic regulation into a unified learning objective, we design a hierarchical reward model that provides structured feedback. This model aligns with the overarching goal of system-wide traffic optimization, focusing on improving throughput, reducing average travel time and enhancing travel-time reliability.

6.2.1 Traffic Pressure and State Representation

Traffic pressure has emerged as a critical metric for quantifying the imbalance between incoming and outgoing traffic at an intersection [Wei et al., 2019a]. Unlike conventional metrics such as

queue length or average delay, which primarily capture localized congestion, *traffic pressure* provides a more dynamic and system-aware representation of congestion by incorporating both demand (entering vehicles) and supply (exiting vehicles). This concept aligns with fundamental traffic flow management and is particularly relevant in RL-based traffic control, where capturing real-time network-wide dynamics is essential for optimal decision-making.

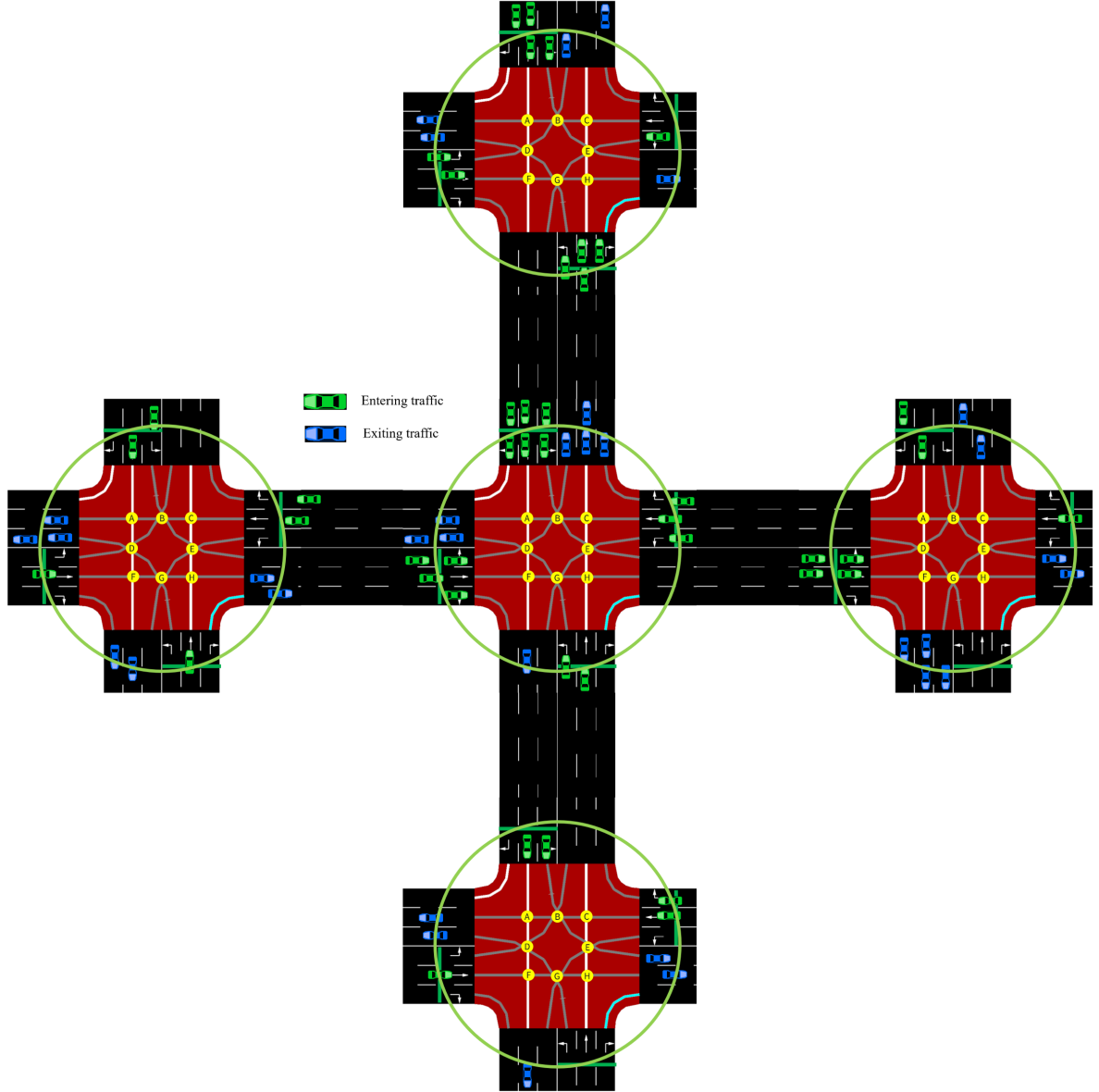


Figure 6.2: The definition of intersection traffic pressure. The green circle at each intersection serves as a trigger for initiating intersection scheduling, when the incoming platoons are determined incompatible.

The pressure of an intersection, as defined in [Chen et al., 2020], is computed as the difference between the sum of queueing vehicles on all entering lanes and the sum of queueing vehicles on all exiting lanes. Specifically, as illustrated in Figure 6.2, the pressure for the central intersection is computed as follows:

$$\begin{aligned}
 \text{Pressure} &= \left| \sum \text{in} - \sum \text{out} \right| \\
 &= |(4 + 6 + 3 + 0) - (3 + 5 + 0 + 1)| \\
 &= 6
 \end{aligned}$$

where Q_{in} represents the number of queueing vehicles on all incoming lanes, and Q_{out} denotes the number of queueing vehicles on all outgoing lanes.

From a system-wide perspective, traffic pressure serves as an essential indicator for proactive congestion mitigation strategies. By dynamically capturing the imbalance at intersections, it enables adaptive traffic signal control policies that contribute to improved network-level performance. RL-based approaches leveraging pressure-based reward functions have demonstrated superior efficiency in reducing overall congestion and optimizing throughput in conventional urban traffic control systems [Wei et al., 2019b, Zhao et al., 2020, Zhang et al., 2022, Wu et al., 2021b].

6.2.2 Lane-Level Traffic Pressure Representation

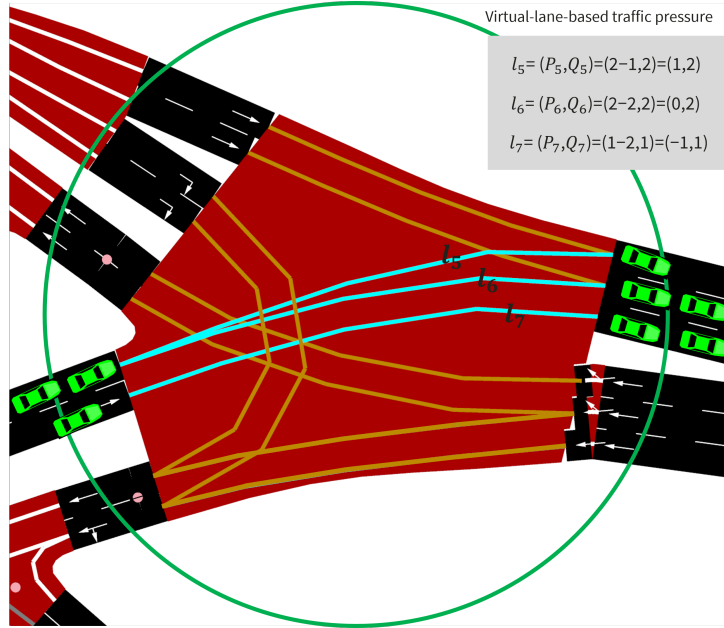


Figure 6.3: A real intersection in Dublin city contains multiple virtual lanes. This example shows how sub-lane-based traffic pressure works at a deformed intersection.

To enhance the state representation, we introduce the concept of a *virtual lane*. A virtual lane is defined relative to an incoming physical lane based on its connections within the intersection. Specifically:

- If a physical lane connects to multiple outgoing directions (e.g., left-turn, straight, or right-turn), each of these paths is represented as a separate virtual lane. A virtual lane thus

represents the specific pairing of an incoming lane with a distinct outgoing lane across the intersection.

- A physical lane allowing vehicles to go straight or turn left will be represented as two virtual lanes, one for the straight connection and one for the left-turn connection. Similarly, if a physical lane connects to three outgoing lanes (e.g., left-turn, straight, and right-turn), it will be represented as three virtual lanes.
- The crossing of a 50-meter radius circle centred at the intersection serves as a trigger for crossing CAVs, allowing the intersection scheduling agent to initiate lane subdivision and intersection scheduling, but only when the incoming platoons in the virtual lanes are determined to be incompatible.

To characterize traffic demand and the traffic difference between incoming and outgoing virtual lanes, each virtual lane i is represented by a tuple (P_i, Q_i) , defined as follows:

- P_i : The *traffic pressure* of virtual lane i , calculated as the queuing vehicle difference between the ego lane and its specific outgoing connection within the intersection.
- Q_i : The *queue length* of virtual lane i , representing the number of vehicles in the incoming virtual lane as demand.

An example of lane-level traffic pressure is provided in Figure 6.3. As depicted, the incoming physical lanes are represented as eleven virtual lanes based on outgoing connections, where one physical lane is split into two virtual lanes leading to different exits, i.e., lanes l_5 and l_6 are considered two virtual lanes in the pressure calculation, while lane l_7 is a virtual lane that leads to a single outgoing direction.

6.2.3 Intersection State Representation

The individual state S of the intersection is formalised as:

$$S = \{(P_1, Q_1), (P_2, Q_2), \dots, (P_N, Q_N)\}$$

where C denotes the total number of virtual lanes at the intersection. Each tuple (P_i, Q_i) represents the traffic pressure and queue length for a specific virtual lane i . The number C depends on the structure of the intersection and the number of cross-intersection connections for each physical lane.

To facilitate multi-agent coordination, the full observation of the traffic optimization agent at intersection i allows it to receive observations from neighbouring intersections, which is formalized as:

$$S_i = \{S_1 || S_2 || \dots || S_K\}, \quad i \in (1, K)$$

where $||$ denotes the concatenate operation and K denotes the total number of directly connected intersections.

6.2.4 Action Space

To effectively manage network traffic, the high-level policy is projected onto each virtual lane, requiring strategic grouping of incoming traffic into platoons based on the optimization objective. The decision-making process for platoon formation is crucial, as it directly impacts traffic flow shaping and intersection throughput.

To achieve this, each virtual lane must determine an action that selects the optimal number of vehicles for platooning. Our approach employs an C -head neural network structure, where each head independently decides the number of vehicles permitted to cross a specific virtual lane i . The decision process is constrained by a masking mechanism, ensuring that the selected number of crossing vehicles remains within feasible limits—bounded by the current demand Q_i and the maximum allowed platoon length L .

The constraint on L is particularly significant, as it defines the upper limit on the number of vehicles dispatched from a lane in a single action, effectively regulating traffic dispersion and preventing excessive congestion. By structuring decisions within these bounds, the model is designed to balance a trade-off between maximizing intersection efficiency and maintaining traffic stability.

For each lane i , the decision on the number of crossing vehicles can be expressed as:

$$\text{Number of crossing vehicles}_i = \min(Q_i, L)$$

The total action space is determined by C , denoted as the number of virtual lanes being managed. Each lane's decision is independent, resulting in a combinatorial action space with L possible choices for each of the C lanes. Therefore, the maximum size of the action space is expressed as:

$$\text{Action Space} = L^C$$

This exponential growth highlights the complexity of managing multiple lanes simultaneously, as the action space increases with both the number of virtual lanes C and the maximum platoon size L . However, by leveraging the multi-head neural network architecture, each head can independently focus on optimising the decision for its respective lane, enabling scalable and efficient decision-making for complex traffic control scenarios.

6.2.5 Reward Model

Designing an effective reward structure in hierarchical reinforcement learning (HRL) presents several key challenges. First, **sparse rewards** are a common issue, particularly for high-level policies, due to the extended temporal nature of their tasks. This sparsity can hinder learning, as agents often receive delayed and infrequent feedback, making it difficult to align actions with desired outcomes. Intrinsic motivation mechanisms, such as curiosity-driven rewards [Pathak et al., 2017], have been proposed to address this challenge by providing denser signals that encourage exploration. Second, **failure responsibility** is critical in HRL, as properly attributing failure signals from the lower layer to the upper layer is essential for effective learning and policy refinement. Accurately crediting

failure signals ensures that the high-level policy can adjust its decisions based on low-level execution errors, thereby improving overall system performance [Bacon et al., 2017]. Third, **reward hacking** [Pan et al., 2022] is a potential risk when reward functions are poorly designed. Agents may exploit training loopholes to achieve high rewards without fulfilling the intended objectives, leading to suboptimal or unintended behaviours [Amodei et al., 2016]. Therefore, careful reward design is essential to ensure alignment between the agent's behaviour and the overall system goals.

Hierarchical Reward Design

To address the multi-objective nature of intersection control, we propose a hierarchical reward formulation that combines the original traffic pressure reward used in PressLight [Wei et al., 2019a] with a solution-quality-based reward from the intersection scheduling agent (see Section 5.2.3).

The overall reward R is defined as:

$$R = \hat{R}_P + \hat{R}_{LBS}$$

where

$$R_P = - \sum_{i=1}^C P_i$$

denotes the negative sum of sub-lane pressures across all incoming lanes at an intersection, representing traffic congestion. The term $R_{LBS} \in (0, 1)$ captures the solution quality of the learned scheduling policy, which evaluates how efficiently vehicles are scheduled to cross the intersection. Higher values correspond to shorter average crossing times relative to a predefined threshold.

To balance the contribution of each component, both R_P and R_{LBS} are normalized using z-score normalization within each training batch:

$$\hat{R}_k = \frac{R_k - \mu_k}{\sigma_k}, \quad k \in \{P, LBS\}$$

where μ_k and σ_k are the mean and standard deviation of the respective reward component over the batch. This normalization ensures consistent gradient magnitudes and prevents domination by either reward term during learning.

To ensure cooperative decision-making in a multi-agent traffic control system, the final reward of an agent i is adjusted to incorporate information from its directly connected neighbouring intersections. The neighbouring reward function is defined as:

$$R'_i = \alpha R_i + (1 - \alpha) \frac{1}{|\mathcal{K}_i|} \sum_{j \in \mathcal{K}_i} R_j$$

where:

- R_i is the immediate reward of intersection i .
- $\mathcal{K}_i$ represents the set of directly connected neighbouring intersections.
- $|\mathcal{K}_i|$ is the number of directly connected intersections.

- R_j is the reward received from a neighbouring intersection j .

This formulation ensures that each intersection agent not only optimizes for its local traffic conditions but also aligns its decision-making with the broader network dynamics. By incorporating the mean reward of its neighbouring intersections, each agent can achieve better coordination and mitigate localized congestion effects.

The proposed hierarchical reward model effectively balances long-term and short-term objectives in HRL by addressing challenges such as sparse rewards, failure attribution, and reward hacking. By incorporating the intersection scheduling agent's solution quality into the reward structure, the traffic network optimization agent receives more frequent reward signals through fast PNMCTS inferences. The model explicitly integrates the decision quality of the intersection scheduling agent using R_{LBS} , ensuring that failure signals are accurately reflected in high-level learning, preventing ambiguous failure attribution and enabling better policy adjustments. Additionally, by averaging rewards across neighbouring intersections, the model discourages egoistic strategies that reduce local traffic pressure at the cost of overall congestion. This cooperative approach aligns individual agent goals with system-wide traffic management, minimizing unintended behaviours. An ablation study on the hierarchical reward and the impact of neighbouring rewards as a function of the cooperation ratio is presented in Section 8.3.4.

6.3 Training Method

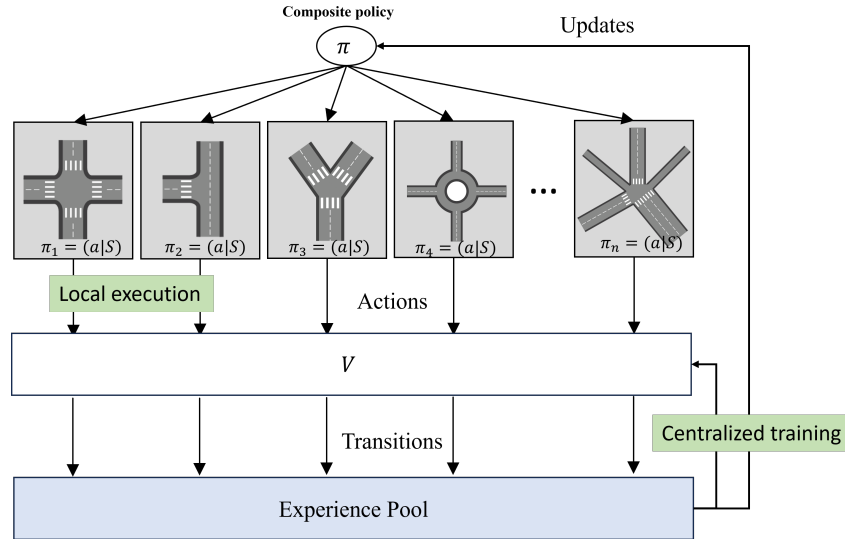


Figure 6.4: Centralised Training with Decentralised Execution (CTDE) framework.

This section introduces the CTDE framework for managing unsignalized urban traffic networks, as depicted in Figure 6.4. During centralised training, agents collaboratively optimise their policies using MAPPO, incorporating both local intersection states and neighbouring traffic pressures. A global value network, i.e., critic, supports policy updates by evaluating overall network conditions. In the decentralised execution phase, agents independently deploy their learned policies, relying solely on local observations. This structure ensures scalable, coordinated traffic management, enhancing throughput and reducing congestion in complex urban environments.

6.3.1 Neural Network Structure and Optimization Strategy

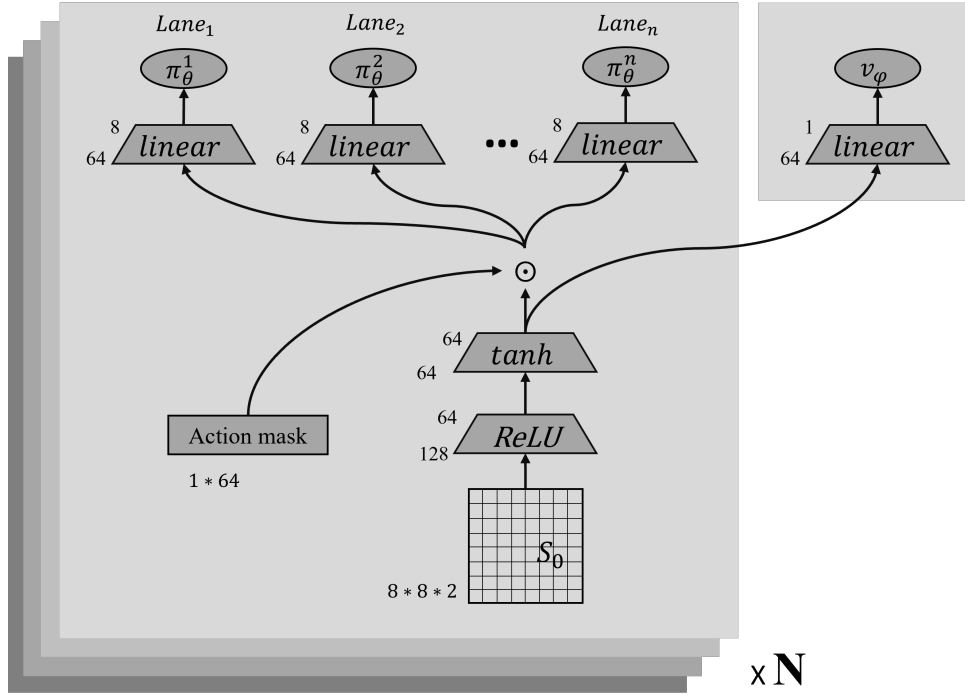


Figure 6.5: High-level neural network structure.

The neural network architecture, designed to optimise scalability and coordination central to our approach, is depicted in Figure 6.5. The action mask is concatenated with the observation and exclusively pass to the policy network for optimization. In contrast, the value network only processes the original state representation without the action mask. Specifically, the structure consists of:

- **A Shared Global Value Network:** A single value network v_ϕ is shared across all agents to estimate the global state value. This design enables the centralised training process to capture the overall traffic network dynamics, allowing agents to collectively optimise for global traffic flow.
- **Independent Policy Networks:** Each agent i maintains an independent policy network π_θ^i , parametrized individually for each virtual lane. These policy networks are optimised to make decentralised decisions during execution while leveraging the global value network during training. This separation ensures that each policy is tailored to a specific intersection while remaining aligned with the network-wide objective.
- **Parallel Action generation:** Each head in the multi-head design corresponds to a specific virtual lane and decides the number of vehicles to be allocated to each subsequent platoon. This parallelism enables the network to process multiple lanes simultaneously, regardless of intersection geometric layout, significantly enhancing computational efficiency.

6.3.2 Training Schema

Centralised Training Phase

The system-wide optimisation combining with intersection scheduling reward is formulated as an objective function, defined as:

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^T \gamma^t R_t \right], \quad (6.1)$$

where θ represents the policy parameters, γ is the discount factor, and R_t is a cooperative reward incorporating both intersection scheduling reward and global traffic pressure reward. To facilitate coordination, each agent receives observations integrating local traffic states with neighbouring pressures, enabling informed, context-aware decision-making.

Training is stabilized through MAPPO, an extension of PPO tailored for decentralized multi-agent learning. MAPPO leverages a shared critic to estimate a global value function, providing each agent with a more accurate advantage function while preserving decentralized execution. This shared critic reduces variance in gradient updates and improves sample efficiency by incorporating experiences from multiple agents into a shared replay buffer. Each agent employs an C -head policy network, where each head generates lane-specific control actions, enabling fine-grained traffic regulation. By using MAPPO, agents achieve stable convergence, effective credit assignment, and improved coordination, ensuring scalability to large, complex traffic networks.

Decentralised Execution Phase

After training, the system enters the decentralised execution phase. In this phase, each agent makes decisions independently, using only its local observations and its trained policy network. This decentralised design improves scalability, allowing the system to manage large urban traffic networks. Although agents act independently, the cooperative learning during training helps them maintain coordinated behaviours. In conclusion, the CTDE-style MAPPO learning framework offers several advantages:

- **Scalability:** The shared global value network enables agents to learn from a unified global objective, eliminating redundant state evaluations across agents. This design ensures that computational complexity does not increase linearly with the number of intersections, making the approach efficient for large-scale deployments. [Kim and Sung, 2023]
- **Coordination with Independence:** While each agent operates independently using its own policy network, these policies are trained within a shared framework that aligns individual decisions with the broader network-wide optimization goal. This setup allows agents to adapt to local conditions while collectively contributing to global traffic efficiency, maintaining a balance between decentralization and coordination. [Yi et al., 2022]
- **Efficient Credit Assignment:** The global value network provides a consistent and holistic evaluation of the overall traffic state. By utilizing this shared value estimation, individual agents receive accurate feedback on their contributions to traffic optimization, enabling

effective credit assignment and reducing the risk of misattributing rewards—a common challenge in multi-agent reinforcement learning. [Zhou et al., 2020]

- **Optimization Stability:** Separating value estimation from policy optimization mitigates interference between learning objectives. This decoupling prevents policy updates from being overly influenced by fluctuating value estimates, leading to more stable training dynamics and improved convergence properties. [Zhou et al., 2020]

6.4 Summary

This chapter addressed the research question: "**What are effective strategies for training hierarchical multi-agent systems for large-scale, heterogeneous urban networks?**" by tackling two key challenges: 1) handling heterogeneous intersection geometries and 2) ensuring coordinated learning and decision-making across multiple intersections.

The first challenge—heterogeneity—arises from variations in intersection layouts, connectivity, and traffic patterns. To address this, the proposed GAL framework transforms different intersection layouts into a standardized board representation using the DTSSM. This uniform abstraction enables a single policy to generalize across diverse geometries without requiring layout-specific retraining. The GAL framework supports both zero-shot generalization—allowing policies to be directly applied to unseen intersections—and fine-tuning for efficient adaptation to complex layouts, significantly enhancing the scalability of the approach.

The second challenge—multi-agent coordination—is critical since intersection-level decisions influence downstream traffic conditions. To ensure cooperative learning, the framework employs MAPPO within a CTDE paradigm, where a global value network provides a shared evaluation function. This allows agents to optimize local policies while considering network-wide effects. Neighbouring intersections are coordinated through the reward function, which incorporates cooperative reward sharing. This mechanism ensures that agents make decisions that benefit overall traffic efficiency rather than optimizing for local performance alone.

To align local decisions with system-wide efficiency, the thesis introduces a hierarchical reward model. Instead of relying solely on localized feedback, the reward function integrates both intersection-level performance and cooperative reward sharing with neighbouring intersections. Additionally, the intersection scheduling layer provides structured feedback, ensuring that high-level decisions lead to long-term network improvements.

In conclusion, the proposed hierarchical MARL framework effectively trains agents for large-scale, heterogeneous urban networks. The hierarchical structure ensures self alignment between layers, the GAL framework enables policy generalization across varying geometries, and the MAPPO-based coordination mechanism fosters cooperative learning. Together, these elements form a structured and scalable reinforcement learning approach for intelligent traffic management.

7 Implementation of Vehicle Trajectory Control in SUMO

This chapter introduces a precise vehicle trajectory calculation method for unsignalized intersection crossing. Given a scheduled crossing from the intersection scheduling agent (see Chapter 5), vehicles must execute trajectories that adhere to assigned time slots, distance constraints, acceleration/deceleration limitations and speed requirements.

To address this, a **multi-segment acceleration control algorithm** is proposed, enabling vehicles to dynamically adjust speed and acceleration throughout the crossing process. The method divides the crossing schedule into a set of **Planned Points** (PPs) corresponding to critical waypoints, such as collision area entrances and exits. Each PP is preceded by a **time slot** from the crossing schedule, during which vehicles adjust their motion to arrive at the PP as required. The baseline assumption considers a constant-speed trajectory; however, real-world conditions often require adjustments to avoid excessive acceleration or deceleration. Therefore, for each PP section, a **two-segment speed control model** is introduced, allowing controlled acceleration adjustments within predefined motion constraints.

A key feature of this approach is its ability to handle both single-vehicle trajectories and platoon-based motion, where an entire platoon is treated as an extended vehicle. This ensures that all vehicles within a platoon maintain synchronization while following the trajectory of the leading vehicle. Furthermore, mathematical formulations and proofs establish feasible acceleration and time constraints, ensuring that vehicles can effectively adapt to intersection schedules while maintaining comfort and safety.

7.1 CAV Parametrization for Unsignalized Urban Intersections

All simulated vehicles in this study are modelled as fully-automated Connected and Automated Vehicles (CAVs) using SUMO's Intelligent Driver Model (IDM) [Albeaik et al., 2022] car-following model, following the parametrization approach of [Gu  riaux and Dusparic, 2020]. This section details the vType configuration employed in our experiments and provides justification for each parameter choice in the context of *unsignalized urban intersections*. It should be emphasized that these settings are implementation prerequisites required to run the simulation and are not inherent to the scientific contribution.

```
<vType id="CAV_IDM" vClass="passenger"
```

```

carFollowModel="IDM"
accel="3.0" decel="4.5" emergencyDecel="9.0"
minGap="1.0" tau="0.5"
sigma="0.05" speedDev="0.05"
lcStrategic="0.5"/>

```

- `carFollowModel=IDM`: The IDM car-following model ensures stable gap-keeping behavior and smooth approach dynamics, while allowing vehicles to maintain sufficiently close following distances to form platoons when required. In the context of unsignalized intersections, the crossing scheduler overrides the default IDM behavior to directly control vehicle motion in accordance with the assigned time slots.
- `accel=3.0 m/s2`: Typical urban CAV capability; allows vehicles to reach slot speeds over short approaches without harsh jerk.
- `decel=4.5 m/s2`, `emergencyDecel=9.0 m/s2`: A comfortable braking limit with a clear emergency reserve.
- `minGap=1.0 m`: A conservative standstill buffer is applied for low-speed queues at intersection approaches, increasing storage capacity by allowing more vehicles to queue before the starting line while avoiding reliance on unrealistically small inter-vehicle gaps.
- `tau=0.5 s`: Short time headway consistent with coordinated CAV operation. It increases entry capacity of conflict zones and reduces approach delay.
- `sigma=0.05`: Low imperfection to emulate automated control and make runs repeatable.
- `speedDev=0.05`: This parameter allows a $\pm 5\%$ deviation from the desired speed, thereby preventing perfectly synchronized platoons from exhibiting artificial lock-step behaviour during merging manoeuvres.
- `lcStrategic=0.5`: Moderates lane-change desire upstream of intersections. This reduces late weaving that would otherwise disrupt approach coordination.

These parameters are needed to enabling the simulation to run under urban unsignalized CAV conditions, but they do not influence the operation of the intersection scheduler or the proposed trajectory-planning method. The scheduler and the multi-segment controller fully determine vehicle speeds and arrival times, while the IDM serves only as the baseline longitudinal control model when explicit control commands are not applied.

7.2 Vehicle Speed Control Problem for Unsignalized Intersection Crossing

Once the intersection scheduling agent generates a compatible schedule for a set of platoons, the corresponding low-level vehicle trajectories must be accurately computed to satisfy the assigned travel time, travel distance, and the specified constraints on arrival speed and acceleration. As

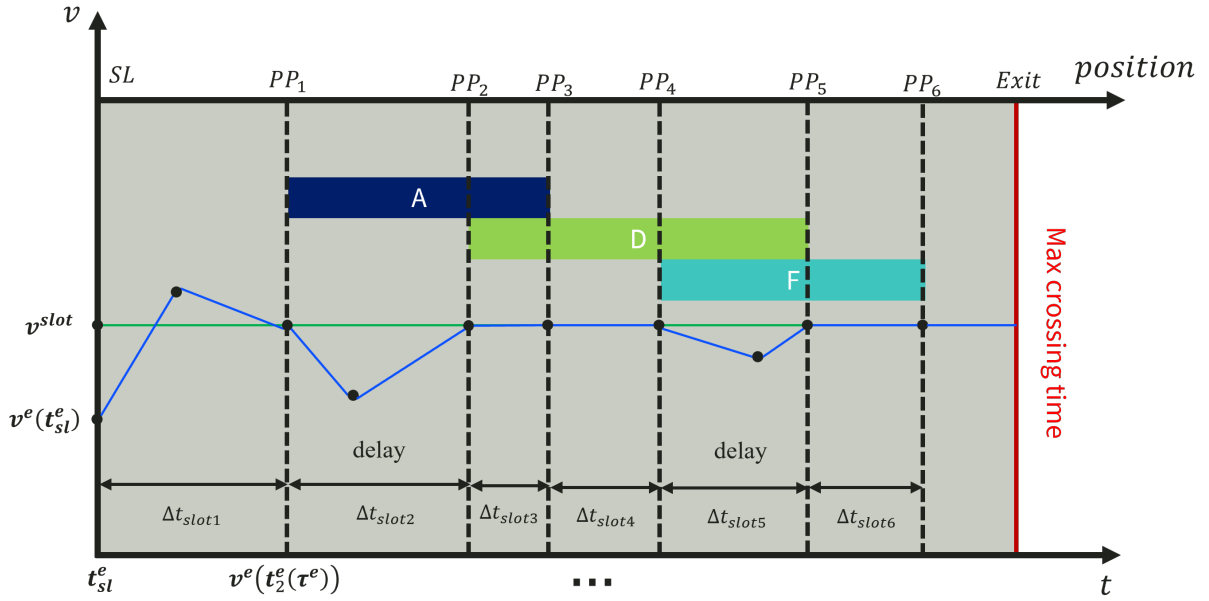


Figure 7.1: The multi-segment trajectory control process.

shown in Figure 7.1, a scheduled trajectory is divided into a series of PPs, each assigned a specific time frame for arrival, referred to as a **time slot**. The PPs reflect the sequence of collision areas traversed during the crossing. For instance, the trajectory illustrated in the figure passes through collision areas A, D, and F.

In this scenario, the first time slot extends from the starting line (SL) to the entrance of A, which is labelled as PP_1 . Similarly, the next time slot goes from the entrance of A to the entrance of B, denoted as PP_2 , while the next time slot from the entrance of B to the exit of A is marked as PP_3 , and so on. The scheduled crossing time slot for each travel segment is provided, such as the time from the SL to PP_1 , represented as Δt_{slot1} .

The baseline trajectory is a flat line (shown as the green line in Figure 7.1), assuming that the vehicle crosses the intersection at a constant speed. However, in most cases, the scheduled crossing requires the vehicle to delay entry into certain collision areas. For example, as shown in the figure, time frames Δt_{slot2} and Δt_{slot5} are extended due to delays. Consequently, the baseline trajectory, which initially follows a constant speed v^{slot} , must be adjusted. When the same distance is travelled over a longer duration, the travel speed decreases, resulting in a concave trajectory, i.e., the trajectory between PP_1 and PP_2 . Notably, **none of the vehicle trajectories involve a complete stop within the intersection midsection**; that is, vehicle speeds remain strictly positive throughout the crossing process. This constraint is explicitly enforced through the design of the intersection scheduling mechanism. Specifically, the action space of the scheduler is dynamically truncated to exclude any scheduling decisions that would result in an intermediate speed less than or equal to zero. Since vehicle speed is a function of both its assigned crossing time and the delay introduced by scheduling decisions, this truncation ensures that only feasible trajectories—those maintaining continuous motion—are considered during action selection. In effect, the bounded delay constraint not only limits the scheduling horizon but also guarantees that all planned trajectories satisfy the minimum speed requirement, thus preventing stop-and-go behaviour within

the intersection.

Conversely, if the initial speed of the ego vehicle, denoted as $v^e(t_{sl}^e)$ at the starting line, is lower than the planned crossing speed v^{slot} (e.g., due to preceding vehicles exiting), the trajectory becomes more aggressive, forming a convex shape to compensate for the delay, i.e., the trajectory between SL and PP_1 .

In conclusion, determining the required speed for each PP can be formulated as a two-segment speed control problem, modelled as a linear motion process with variable constant acceleration. Since the entire crossing is divided into multiple PP sections, the overall process constitutes a multi-segment acceleration control system. A detailed illustration of the acceleration control for a single ego vehicle travelling one PP section is provided in Figure 7.2.

7.2.1 Notation

This section provides precise definitions of the key variables used to describe the motion dynamics of vehicles within the system. The variables are categorized into three groups: position and distance-related variables, time-related variables, speed and acceleration-related variables.

Position and Distance Definitions

- $D_{PP_i}^e$: The distance between the ego vehicle and PP_i .
- $\Delta D^v(p)$: The distance travelled by vehicle v during period p .

Trajectory Definitions

- τ^v : trajectory of vehicle v

Time Definitions

- t_{sl}^v : The time at which vehicle v arrives at the SL.
- Δt_{slot_i} : The time required for vehicle v to arrive at PP_i , $i > 1$, starting from t_{sl}^v .
- $\Delta t_n^v(\tau^v)$: The time interval between $t_{n-1}^v(\tau^v)$ and $t_n^v(\tau^v)$ for $n > 0$.

Speed and Acceleration Definitions

- v^{slot} : The constant baseline speed of the crossing, also serves as the target speed for the crossing vehicle to match when arrive each PP.
- $v^e(t_2^e(\tau^e))$: The velocity of the ego vehicle at the end of a two-segment acceleration trajectory τ^e .
- v_{max} : The maximum allowed velocity of a vehicle.
- $a_1^e(\tau^e)$: The first segment acceleration of the ego vehicle when executing trajectory τ^e .
- a_{max}^v : The maximum comfortable acceleration of vehicle v .

- d_{max}^v : The maximum comfortable deceleration of vehicle v .

7.3 Multi-segment Acceleration Control Algorithm

Since instantaneous speed changes are unrealistic in real-world control, the key challenge is to determine, for each slot, the appropriate timing to adjust the acceleration based on the given dynamics at the ego vehicle's starting position. As illustrated in Figure 7.2, for the first segment, the variables to solve are $a_1^e(\tau^e)$ and $\Delta t_1^e(\tau^e)$, while for the second segment, the variables to solve are $a_2^e(\tau^e)$ and $\Delta t_2^e(\tau^e)$.

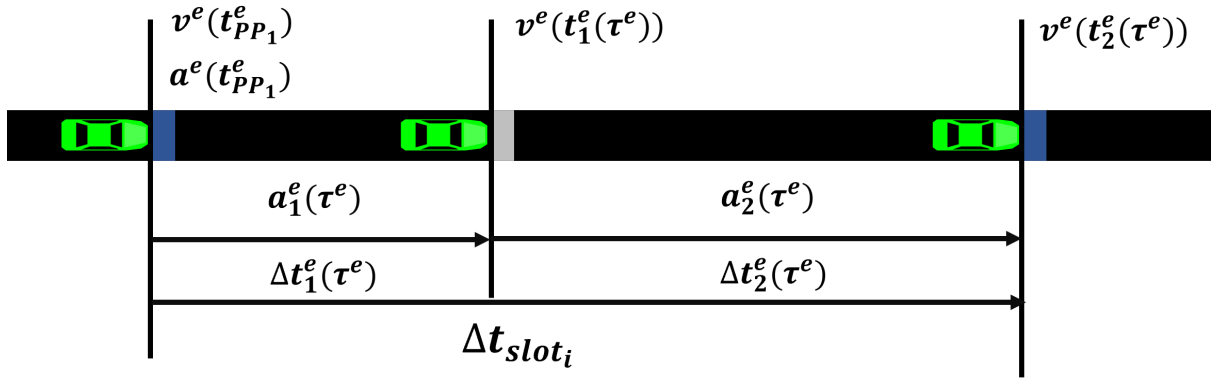


Figure 7.2: Two-segment formulation of the slot synchronization process.

Algorithm 1 Synchronization Process for Ego Vehicle

Input: t_{sl}^e : Arrival time of ego vehicle at SL.

Output: τ^e : Allocated trajectory for the ego vehicle.

1 Initialization:

At time t_{sl}^e , trajectory calculator obtains the following parameters: $D_{PP}^e(t_{sl}^e)$, a_{max}^e , v^{slot} , and $v^e(t_{sl}^e)$.

2 With a given Δt_{slot}

while true do

3 **if** $a_{min} \leq a_1^e(\tau^e) \leq a_{min}$ **and** $a_{min} \leq a_2^e(\tau^e) \leq a_{min}$ **then**

4 | break

5 | Using the given parameters, determine $t_1^e(\tau^e)$, $t_2^e(\tau^e)$, $a_1^e(\tau^e)$, $a_2^e(\tau^e)$, and τ^e

6 Send Trajectory: trajectory calculator sends τ^e to the vehicle

7 Execute Trajectory: The vehicle executes the allocated trajectory τ^e at t_{sl}^e

The basic algorithm applies to a single ego vehicle travelling through one PP section. For a platoon, the same algorithm is used but treats the **entire platoon as a single extended vehicle**. Each vehicle within the platoon is assumed to have identical power and dynamic capabilities, enabling it to follow the trajectory of the leading vehicle throughout the entire crossing process.

Assume there is an intermediate state of the ego vehicle at time $t_1^e(\tau^e)$ with speed $v^e(t_1^e(\tau^e))$ and acceleration $a_1^e(\tau^e)$. The first segment has a duration $\Delta t_1^e(\tau^e)$, and the second segment has a duration $\Delta t_2^e(\tau^e)$.

At time $t_2^e(\tau^e)$, the speed of the ego vehicle equals the target speed, and the acceleration of the ego vehicle needs to reset to zero for next control, such that:

$$v^e(t_2^e(\tau^e)) = v^{\text{slot}} \quad (7.1)$$

$$a_2^e(\tau^e) = 0 \quad (7.2)$$

Note that the total duration of the synchronization process equals the given time slot:

$$\Delta t_{\text{slot}} = \Delta t_1^e(\tau^e) + \Delta t_2^e(\tau^e) \quad (7.3)$$

The total distance for the synchronization process is denoted as $D_{\text{PP}}^e(t_{\text{sl}}^e)$. At t_{sl}^e , the values $D_{\text{PP}}^e(t_{\text{sl}}^e)$, a_{max}^e , v^{slot} , and $v^e(t_{\text{sl}}^e)$ are known. The algorithm executes as follows:

1. The trajectory calculator receives the assigned time slot Δt_{slot} and $D_{\text{PP}}^e(t_{\text{sl}}^e)$. The trajectory calculator selects a value for the first segment's acceleration duration, $\Delta t_1^e(\tau^e)$, based on the given Δt_{slot} , using the inequality provided below:

$$\frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) - v^e(t_{\text{sl}}^e)\Delta t_{\text{slot}} - v^{\text{slot}}\Delta t_{\text{slot}}}{a_{\text{max}}^e\Delta t_{\text{slot}} + v^e(t_{\text{sl}}^e) - v^{\text{slot}}} \leq \Delta t_1^e(\tau^e) \leq \frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) + a_{\text{max}}^e\Delta t_{\text{slot}}^2 - 2v^{\text{slot}}\Delta t_{\text{slot}}}{a_{\text{max}}^e\Delta t_{\text{slot}} + v^e(t_{\text{sl}}^e) - v^{\text{slot}}}. \quad (7.4)$$

2. Having Δt_{slot} and $\Delta t_1^e(\tau^e)$, the trajectory calculator calculates the following parameters:

- (a) Midterm speed:

$$v^e(t_1^e(\tau^e)) = \frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) - v^e(t_{\text{sl}}^e)\Delta t_1^e(\tau^e) - v^{\text{slot}}\Delta t_2^e(\tau^e)}{\Delta t_1^e(\tau^e) + \Delta t_2^e(\tau^e)} \quad (7.5)$$

- (b) Acceleration for the first segment:

$$a_1^e(\tau^e) = \frac{v^e(t_1^e(\tau^e)) - v^e(t_{\text{sl}}^e)}{\Delta t_1^e(\tau^e)} \quad (7.6)$$

- (c) Acceleration for the second segment:

$$a_2^e(\tau^e) = \frac{v^{\text{slot}} - v^e(t_1^e(\tau^e))}{\Delta t_2^e(\tau^e)} \quad (7.7)$$

- (d) Second segment time:

$$\Delta t_2^e(\tau^e) = \Delta t_{\text{slot}} - \Delta t_1^e(\tau^e) \quad (7.8)$$

At $t_0^e(\tau^e)$, the ego vehicle executes the acceleration as $a_1^e(\tau^e)$. After $\Delta t_1^e(\tau^e)$, the ego vehicle sets the acceleration to $a_2^e(\tau^e)$. After $\Delta t_2^e(\tau^e)$, the ego vehicle arrives at the PP.

7.4 Determining the Duration of First-Segment Acceleration

At time t_{sl}^e , five elements are known: the current speed of the ego vehicle, $v^e(t_{sl}^e)$, the target speed of time slot, v^{slot} , the distance between the ego vehicle and the PP, $D_{PP}^e(t_{sl}^e)$, the maximum comfortable acceleration of the ego vehicle, a_{max}^e , and the time to the arrival of the target slot, Δt_{slot} . Based on these parameters, the objective is to solve for $\Delta t_1^e(\tau^e)$. The synchronization process is illustrated in Figure S1.

According to the equation of uniformly accelerated linear motion, the two-segment constant-acceleration linear motion equation is:

$$D_{PP}^e(t_{sl}^e) = \frac{(v^e(t_{sl}^e) + v^e(t_1^e(\tau^e))) \cdot \Delta t_1^e(\tau^e)}{2} + \frac{(v^e(t_1^e(\tau^e)) + v^{slot}) \cdot \Delta t_2^e(\tau^e)}{2}. \quad (1)$$

In Equation (1), $D_{PP}^e(t_{sl}^e)$ and v^{slot} are known, and the intermediate speed $v^e(t_1^e(\tau^e))$ is the variable of interest. This can be rewritten as:

$$v^e(t_1^e(\tau^e)) = \frac{2D_{PP}^e(t_{sl}^e) - v^e(t_{sl}^e)\Delta t_1^e(\tau^e) - v^{slot}\Delta t_2^e(\tau^e)}{\Delta t_1^e(\tau^e) + \Delta t_2^e(\tau^e)}. \quad (2)$$

The total synchronization time is equal to the given time slot to arrive PP:

$$\Delta t_{slot} = \Delta t_1^e(\tau^e) + \Delta t_2^e(\tau^e).$$

Substituting $\Delta t_1^e(\tau^e) + \Delta t_2^e(\tau^e)$ with Δt_{slot} into Equation (2):

$$v^e(t_1^e(\tau^e)) = \frac{2D_{PP}^e(t_{sl}^e) - v^e(t_{sl}^e)\Delta t_1^e(\tau^e) - v^{slot}\Delta t_2^e(\tau^e)}{\Delta t_{slot}}. \quad (3)$$

The accelerations in segments one and two can be expressed as:

$$a_1^e(\tau^e) = \frac{v^e(t_1^e(\tau^e)) - v^e(t_{sl}^e)}{\Delta t_1^e(\tau^e)} \leq a_{max}^e, \quad (4)$$

$$a_2^e(\tau^e) = \frac{v^{slot} - v^e(t_1^e(\tau^e))}{\Delta t_{slot} - \Delta t_1^e(\tau^e)} \leq a_{max}^e. \quad (5)$$

Replacing $v^e(t_1^e(\tau^e))$ in Equation (4) using Equation (3):

$$(v^{slot} - v^e(t_{sl}^e) - a_{max}^e \Delta t_{slot}) \cdot \Delta t_1^e(\tau^e) + (2D_{PP}^e(t_{sl}^e) - v^e(t_{sl}^e)\Delta t_{slot} - v^{slot}\Delta t_{slot}) \leq 0. \quad (6)$$

Similarly, replacing $v^e(t_1^e(\tau^e))$ in Equation (5) using Equation (3):

$$(2v^{slot}\Delta t_{slot} - 2D_{PP}^e(t_{sl}^e) - a_{max}^e \Delta t_{slot}^2) - (v^{slot} - v^e(t_{sl}^e) - a_{max}^e \Delta t_{slot}) \cdot \Delta t_1^e(\tau^e) \leq 0. \quad (7)$$

As for a two-segment acceleration process, the factor $v^{slot} - v^e(t_{sl}^e) - a_{max}^e \Delta t_{slot}$ is always negative

(see Box 1), thus dividing inequalities (6) and (7) by $v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}}$ can be written as follows:

$$\Delta t_1^e(\tau^e) + \frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) - v^e(t_{\text{sl}}^e)\Delta t_{\text{slot}} - v^{\text{slot}}\Delta t_{\text{slot}}}{v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}}} \geq 0, \quad (8)$$

$$\Delta t_1^e(\tau^e) + \frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) + a_{\text{max}}^e \Delta t_{\text{slot}}^2 - 2v^{\text{slot}}\Delta t_{\text{slot}}}{v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}}} \leq 0. \quad (9)$$

Combining Inequalities (8) and (9):

$$\frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) - v^e(t_{\text{sl}}^e)\Delta t_{\text{slot}} - v^{\text{slot}}\Delta t_{\text{slot}}}{a_{\text{max}}^e \Delta t_{\text{slot}} + v^e(t_{\text{sl}}^e) - v^{\text{slot}}} \leq \Delta t_1^e(\tau^e) \leq \frac{2D_{\text{PP}}^e(t_{\text{sl}}^e) + a_{\text{max}}^e \Delta t_{\text{slot}}^2 - 2v^{\text{slot}}\Delta t_{\text{slot}}}{a_{\text{max}}^e \Delta t_{\text{slot}} + v^e(t_{\text{sl}}^e) - v^{\text{slot}}}. \quad (10)$$

Inequality (10) gives the range of $\Delta t_1^e(\tau^e)$ for any given Δt_{slot} .

Box 1: Proving $v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}} < 0$ for a Two-Segment Variable Acceleration Process

Assume $\bar{a}^e$ is the average acceleration of the ego vehicle during the synchronization process:

$$v^{\text{slot}} = v^e(t_{\text{sl}}^e) + \bar{a}^e \Delta t_{\text{slot}}.$$

In a two-segment acceleration process, $\bar{a}^e < a_{\text{max}}^e$, which implies:

$$\bar{a}^e \Delta t_{\text{slot}} < a_{\text{max}}^e \Delta t_{\text{slot}}.$$

Given $v^e(t_{\text{sl}}^e) \geq 0$, we have:

$$v^e(t_{\text{sl}}^e) + \bar{a}^e \Delta t_{\text{slot}} < v^e(t_{\text{sl}}^e) + a_{\text{max}}^e \Delta t_{\text{slot}}.$$

Therefore:

$$v^{\text{slot}} < v^e(t_{\text{sl}}^e) + a_{\text{max}}^e \Delta t_{\text{slot}}.$$

This leads to:

$$v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}} < 0. \quad (11)$$

For a one-segment constant acceleration process, where $\bar{a}^e \leq a_{\text{max}}^e$, it follows that:

$$v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}} \leq 0.$$

Specifically, if $v^{\text{slot}} - v^e(t_{\text{sl}}^e) - a_{\text{max}}^e \Delta t_{\text{slot}} = 0$, the ego vehicle is using a_{max}^e for a constant acceleration process. To avoid a division by zero error in such cases, we manually set:

$$\Delta t_1^e(\tau^e) = \Delta t_2^e(\tau^e) = \frac{1}{2} \Delta t_{\text{slot}}.$$

7.5 Summary

In conclusion, this chapter presents a multi-segment trajectory calculation method that seamlessly integrates with the intersection scheduling agent, ensuring precise and smooth crossings in unsignalized intersections. By modelling vehicle motion as a series of two-segment acceleration control problems, the approach enables real-time speed adjustments to adhere to the scheduling plan while maintaining dynamic constraints. The method applies to both individual CAVs and platoons, ensuring synchronized motion and strict adherence to assigned time slots. Ultimately, this method enhances CAV trajectory execution, ensuring alignment with the intersection scheduling plan for improved traffic coordination.

8 Evaluation

This chapter presents a comprehensive evaluation of the proposed hierarchical, multi-agent reinforcement learning, i.e., hierarchical multi-agent proximal policy optimization (HRL-MAPPO), framework for unsignalized urban traffic network optimization, in alignment with the research questions outlined in this thesis. The evaluation systematically investigates the framework’s performance, robustness, and scalability across a variety of traffic management scenarios, ranging from single unsignalized intersection scheduling, to homogeneous multi-intersection grids, and large-scale heterogeneous urban network optimization. The objective is to validate the framework’s capability in optimizing traffic flow while addressing key challenges in hierarchical decision-making, intersection representation, reward balancing, and large-scale MARL training.

All experiments are conducted using Simulation of Urban Mobility (SUMO) [Lopez et al., 2018b], a widely adopted, open-source microscopic traffic simulator designed for modelling, simulating, and evaluating traffic dynamics at large scale. SUMO provides high-fidelity modelling of vehicle behaviours, supports detailed intersection control, and enables flexible scenario construction across various road network topologies, making it a suitable platform for evaluating intelligent traffic control algorithms in both synthetic and real-world environments.

After we set up baselines and metric in Section 8.1, the evaluation process follows the roadmap illustrated in Figure 8.1. We begin by assessing the performance of the intersection scheduling algorithm in Section 8.2. Specifically, Section 8.2.1 presents results on a single-intersection scenario, comparing the proposed method against a first-in-first-out (FIFO) baseline. Section 8.2.2 extends the evaluation to a homogeneous grid network, where the performance is compared against both traditional and reinforcement learning-based traffic signal control methods. Subsequently, Section 8.2.3 investigates the impact of key design choices and training procedures across four dimensions: 1) evaluation of different neural network architectures, 2) assessment of the proposed parallel neural Monte Carlo Tree Search (PNMCTS) enhancements on the training dataset, 3) generalization performance on the testing dataset, and 4) evaluation under progressively expanding deployment scenarios.

Building on the intersection scheduling evaluation, we advance to system-wide traffic network optimization in Section 8.3. The comparison between the proposed hierarchical framework and state-of-the-art RL methods on single intersections is discussed in Section 8.3.1. Evaluation results on a homogeneous 3×3 grid network, benchmarked against both traditional and RL-based traffic light control methods, are presented in Section 8.3.2.

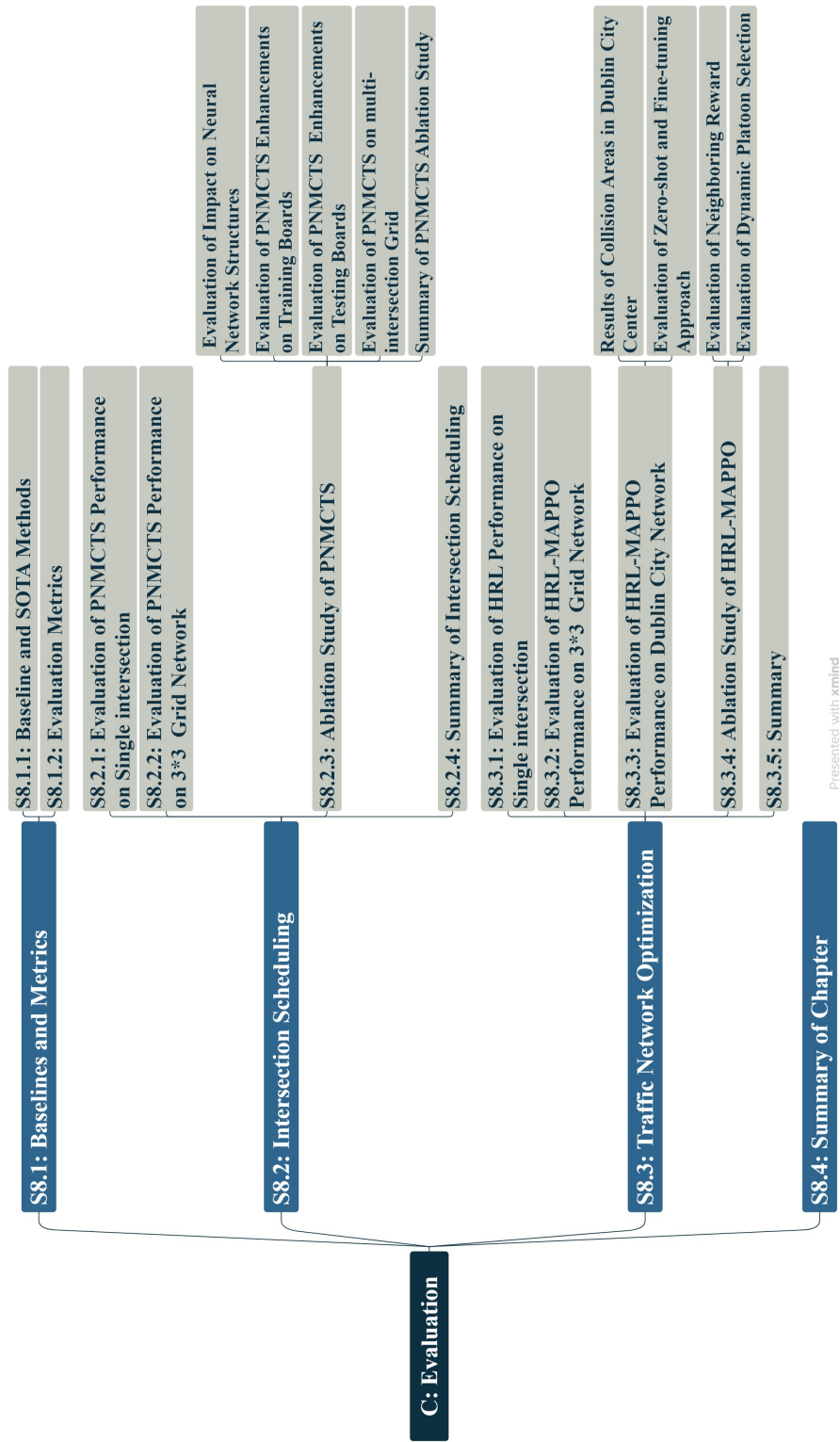


Figure 8.1: Evaluation Roadmap of the Proposed HRL-MAPPO Traffic Network Optimization Framework.

Finally, large-scale evaluation is conducted on a heterogeneous network modelled after Dublin city. Section 8.3.3 presents results comparing the proposed framework under zero-shot and fine-tuned settings. To validate the proposed neighbouring reward design and dynamic platoon selection in HRL-MAPPO, the training details are presented in ablation studies in Section 8.3.4 and Section 8.3.4, respectively.

8.1 Evaluation Baselines and Metrics

This section outlines the baseline methods and evaluation metrics against which the proposed framework is compared across the different scenarios. Additionally, as a representative heterogeneous urban network we use the Dublin city centre road network, comprising 1,645 intersections, to examine the scalability and adaptability of the framework using zero-shot and few-shot fine-tuning techniques across varying intersection types.

8.1.1 Baselines for Single Unsignalised Intersection

For single *unsignalized* intersections, we compare against the methods reviewed in our related work in Section 3.2.3, because they span the main design choices and performance frontiers in this setting. Reservation-based AIM [Dresner and Stone, 2008] is the canonical non-learning baseline. R&B-AIM and O&B-AIM [Yu et al., 2019] extend the reservation idea with batching and optimization, yielding strong throughput under diverse demands. DCL-AIM [Wu et al., 2019b] represents decentralized cooperative learning for conflict resolution. INTEL-PLT [Li et al., 2022] and COOR-PLT [Li et al., 2023a] use learning with platoon/priority structures and report state-of-the-art delay and safety metrics at a single intersection. Together, these methods cover (i) classic reservation control, (ii) batched/optimized variants, and (iii) modern learning-based controllers, providing a broad, fair, and well-documented baseline set consistent with our related-work analysis.

All these baselines were evaluated on a four-way intersection with three lanes per approach, and their reported results are taken directly from the COOR-PLT paper. To ensure a fair comparison, our experiments use the same traffic flow and intersection geometry settings as those reported in that paper.

Reservation-based AIM and R&B-based AIM both implement a First-Come-First-Serve (FCFS) policy for handling reservations from connected and autonomous vehicles (CAVs).

Reservation-based AIM focuses on individual CAV reservations, whereas R&B-based AIM coordinates fixed-size platoons or batches of CAVs, which helps to streamline intersection flow.

O&B-based AIM similarly applies fixed-size platooning as in R&B-based AIM, but instead optimises the sequence in which platoons are serviced to reduce overall traffic delays, as opposed to the straightforward FCFS approach.

DCL-AIM and INTEL-PLT are both methods underpinned by reinforcement learning (RL).

DCL-AIM applies a coordination strategy focused on individual CAVs, while INTEL-PLT adopts an adaptive platooning method that adjusts the platoon configuration based on traffic conditions.

COOR-PLT utilises a hierarchical approach with dynamically adjustable platoon lengths, aiming to minimise travel time across an intersection.

8.1.2 Baselines for Homogeneous Multi-Intersections

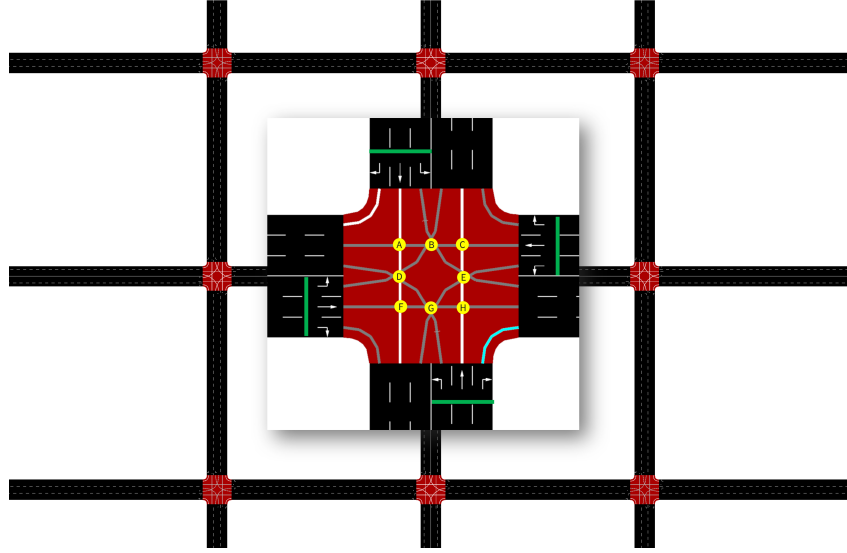


Figure 8.2: A homogeneous 3×3 grid intersection network.

In the homogeneous 3×3 grid network shown in Figure 8.2, we evaluate the proposed method against several established traffic signal control systems, including fixed-time control [Webster, 1958], actuated control [Lopez et al., 2018a], and the state-of-the-art RL-based controller PressLight [Wei et al., 2019a]. For completeness, we note that one RL-based traffic control method designed for unsignalized intersection [Xu et al., 2021b] is excluded from comparison because its implementation is not publicly available and it has not been evaluated on the specific four-way, three-lane intersection geometry considered in our study.

We further introduce two hierarchical RL baselines for comparison: Isolated PPO (ISO-PPO) and Multi-Agent PPO (MAPPO). In ISO-PPO, each intersection operates an independent HRL-PPO agent without inter-agent communication, whereas MAPPO represents a coordinated variant (HRL-MAPPO) enabling joint policy optimization across intersections.

These baselines capture distinct traffic control paradigms, providing standardized and reproducible benchmarks for assessing the adaptability and effectiveness of our method in a structured and uniform grid environment.

8.1.3 Baselines for Heterogeneous Urban Networks

For the Dublin city centre network, which consists of 1,645 heterogeneous intersections, we compare the actuated traffic light control method with our proposed **zero-shot** and **few-shot fine-tuning** techniques. The actuated traffic light control serves as a strong rule-based baseline, where signal phases are adjusted in real time based on detected traffic demand using sensors or loop detectors. In this setup, agents are deployed across 1,500 intersections, where each agent maps intersection dynamics onto an 8×8 grid layout. The remaining intersections, which exceed eight

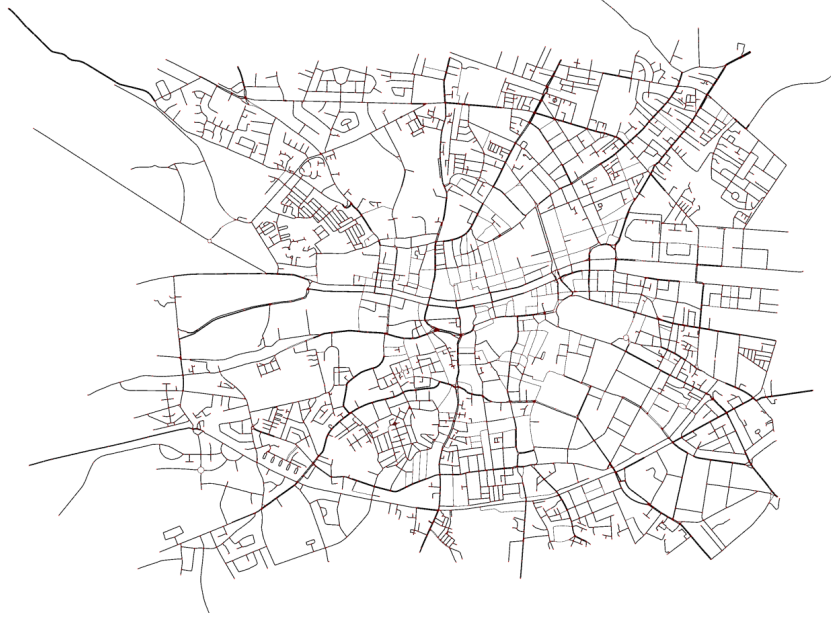


Figure 8.3: Dublin city centre traffic network.

virtual lanes, operate under the default actuated traffic light control. Among the agent-controlled intersections, 200 are selected on the heavy congested neighbourhood for fine-tuning. This selection process evaluates the agents' ability to generalize across diverse intersection types and optimize traffic flow within a complex urban network.

8.1.4 Metrics

To rigorously evaluate the effectiveness of our framework, we adopt a set of standard traffic performance metrics, including: total throughput (TT), average travel time (ATT), and travel-time reliability (TTR). Specifically:

- **Total Throughput (TT):** Defined as the total number of completed trips within simulation period, reflecting the system's overall capacity to process traffic.
- **Average Travel Time (ATT):** Computed as the ratio of the total travel time to the number of completed trips. This metric captures the mean duration a vehicle spends traversing the network from origin to destination.
- **Travel-Time Reliability (TTR):** Measures the variability in travel time across different trips. It reflects the system's consistency and predictability.

The evaluation metrics selected for both single and multiple intersection scenarios are TTP, ATT, and TTR [Federal Highway Administration, 2006]. SUMO inherently supports measurements for TT and ATT, while TTR is not directly provided. Therefore, we must derive TTR from the available data output by SUMO to calculate TTR accurately. The **planning time index (PTI)** [Federal Highway Administration, 2006] is a metric used to quantify the reliability of travel times on a roadway segment. It is defined as the ratio of the 95th percentile travel time to the free-flow travel time. The equation for PTI is given by:

$$PTI = \frac{T_{95}}{T_{\text{free-flow}}}$$

where:

- T_{95} is the 95th percentile travel time, meaning the travel time that is exceeded only 5% of the time.
- $T_{\text{free-flow}}$ is the free-flow travel time, representing the travel time under free-flow conditions.

The PTI indicates how much extra time a traveller should budget to ensure on-time arrival 95% of the time compared to the free-flow travel time. **A higher PTI suggests less reliability in travel times, indicating that travellers need to plan for significantly longer trips to ensure punctuality.** For example, a PTI of 1.60 suggests that travellers should anticipate travel times to be 60% longer than the free-flow travel time to ensure on-time arrival 95% of the time.

8.2 Evaluation of the Intersection Scheduling Method

A series of experiments were conducted to evaluate intersection scheduling performance under varying traffic and network conditions. These experiments span scenarios ranging from single intersections to 3×3 homogeneous intersection networks. The rationale behind the design choices is supported by a comprehensive set of ablation studies, which systematically examine the impact of neural network architectures and enhancements to the standard NMCTS algorithm under both empty and busy intersection. These studies provide empirical evidence supporting the effectiveness of the proposed PNMCTS algorithm in intersection management. Furthermore, the influence of agent deployment rates is investigated to assess the scalability and robustness of the proposed approach.

8.2.1 Evaluation of PNMCTS Performance on Single intersection

To further assess its effectiveness, we compare **PNMCTS-RECS**, i.e., the variant that integrates all proposed enhancements—namely best-so-far Resampling, Entropy regulation, Curriculum learning, and short-path Search, with the FIFO scheduling strategy [Ba and Tordeux, 2023]. FIFO serves as a basic traffic control baseline for unsignalized intersections, where vehicles are allowed to cross strictly in the order of arrival. This rule treats the intersection as a single queue, granting right-of-way to the earliest arriving vehicle regardless of its direction. In scenarios involving schedule conflicts, newly arriving vehicles are delayed until the current crossing vehicle has cleared the intersection. We conducted experiments on 100 test boards under two distinct traffic conditions: *empty boards*, representing intersections with no vehicles occupied, and *busy boards*, representing intersections with undergoing schedules, as illustrated in Figure 8.4.

FIFO serves as a baseline traffic control strategy for unsignalized intersections, where vehicles cross in the order of their arrival. It treats the intersection as a queue: the earliest arriving vehicle is granted priority, followed sequentially by others. In scenarios with overlapping schedules, incoming platoons must wait until the current schedule is entirely completed before accessing the intersection.

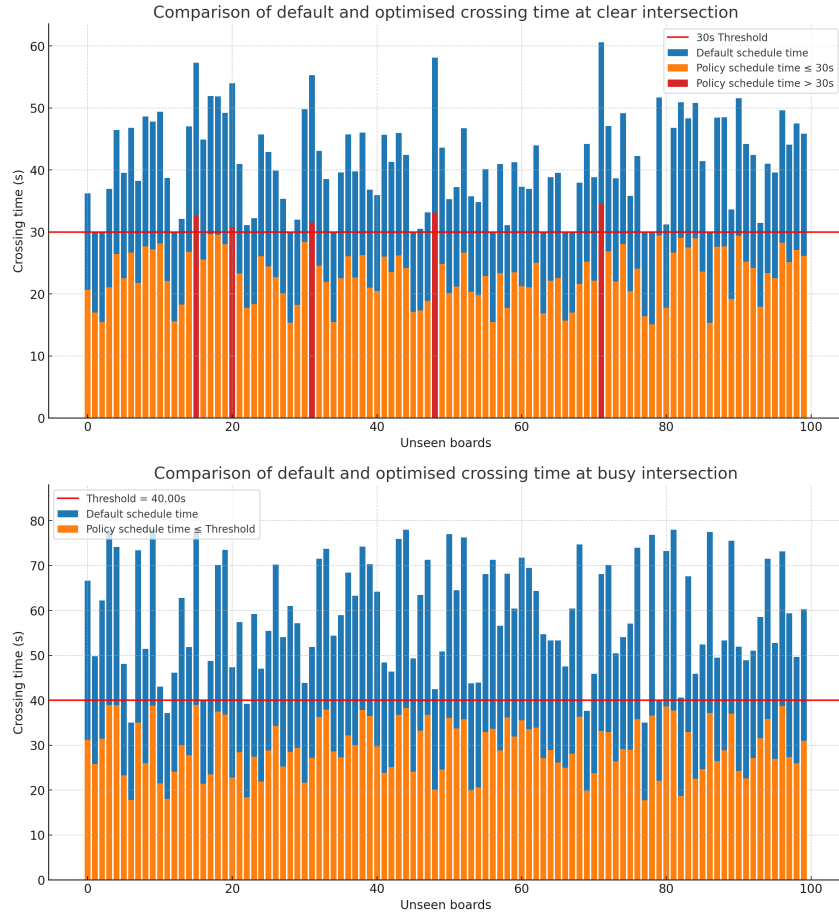


Figure 8.4: Crossing time comparison between FIFO and PNMCTS method at, (a) Low-traffic intersections and (b) High-traffic intersections. All solutions provided by PNMCTS outperforms the FIFO in both scenarios.

Under empty boards (Figure 8.4(a)), our method successfully completed 95 out of 100 schedules within the 30-second crossing threshold, achieving a 43% reduction in average crossing time compared to FIFO. In busy boards (Figure 8.4(b)), where a relaxed threshold of 40 seconds was applied to account for overlapping boards complicity, our approach completed all schedules while reducing the average crossing time by 52% relative to FIFO.

8.2.2 Evaluation of PNMCTS Performance on 3×3 Grid Network

We conducted experiments on a 3×3 grid network to evaluate the performance of the proposed PNMCTS method against traditional and state-of-the-art traffic control strategies. The experiments were performed under the same conditions, with a vehicle travelling speed of 10 m/s and traffic volume increasing from 300 vehicles per hour per lane (veh/h/lane) to 700 veh/h/lane to align with the baseline setup. In SUMO, traffic flows are defined in preprogrammed input files, making simulations deterministic by default. For fixed-time signals, which follow a static schedule, repeated runs always produce the same results. Actuated signals adapt to vehicle presence, but if vehicle routes and departure times are fixed and no stochastic elements (e.g., random insertion or routing) are introduced, the simulation remains deterministic. Therefore, both fixed-time and actuated signal scenarios yield consistent results across runs. The comparison results are summarized in Table 8.1.

Table 8.1: Travel time and throughput comparison in a 3×3 grid network with 3 runs

Metrics	PNMCTS	Fixed-time	Actuated	LIT	PressLight
ATT (s)	22.25 ± 1.7	663.07 ± 0	278.45 ± 0	183.21 ± 5.1	87.51 ± 3.8
TT (t)	4936 ± 15	1024 ± 2	3203 ± 5	3920 ± 6	4251 ± 12

The results clearly demonstrate that the proposed **PNMCTS** method significantly outperforms both traditional and state-of-the-art traffic control methods. Specifically, PNMCTS achieves an ATT of **22.25 seconds**, representing a reduction of:

- **96.6%** compared to traditional fixed-time control (663.07 s),
- **92.0%** compared to the actuated control method (278.45 s),
- **87.9%** compared to the LIT method (183.21 s), and
- **74.5%** compared to the PressLight method (87.51 s).

In terms of TT, PNMCTS achieves **4936** of completed trips, which represents an improvement of:

- **381.9%** over fixed-time control (1024 trips),
- **54.1%** over actuated control (3203 trips),
- **25.9%** over the LIT method (3920 trips), and
- **16.1%** over the PressLight method (4251 trips).

The significant reduction in travel time and improvement in total throughput highlight the effectiveness of PNMCTS in optimizing traffic flow.

8.2.3 Ablation Experiments

This section provides a detailed analysis of 1) impact of different neural network (NN) structures and 2) the four enhancements introduced by the proposed modifications to PNMCTS for solving proposed scheduling board games in Section 5.1.2. We conducted an ablation study to evaluate the impact of each modification against the baseline implementation of standard NMCTS. The four proposed modifications are as follows:

1. **Best-so-far (BSF) resampling:** Reintroduction of the best-so-far trajectory into training examples to improve policy stability.
2. **Entropy Loss Integration:** Incorporating an entropy regulation term into the loss function to improve exploration.
3. **Curriculum Learning:** Gradual learning starting from simple board configurations and incrementally introducing more complex boards (busy boards). Specifically, 10% of the simple boards are replaced with busy boards after every 10k training steps to guide the policy towards generalisable strategies.
4. **Short Path Search (SPS):** An additional short MCTS rollout during evaluation to stabilise performance on complex boards while maintaining reasonable inference time.

Evaluation of Impact on Neural Network Structures

Three neural network architectures are trained and evaluated: a shared network, separate value and policy networks, and a larger shared network. Each architecture is trained on 400 synthetic busy boards for 250k steps in Figure 8.5. Each stored policy is then evaluated on 100 synthetic busy boards, with three evaluation runs per board. The mean win rate, defined as the ratio of solving a board in 30 seconds crossing time contains, along with the mean of schedule quality, i.e., shorter crossing time contributes to higher schedule quality, and mean inference time (in seconds), is reported.

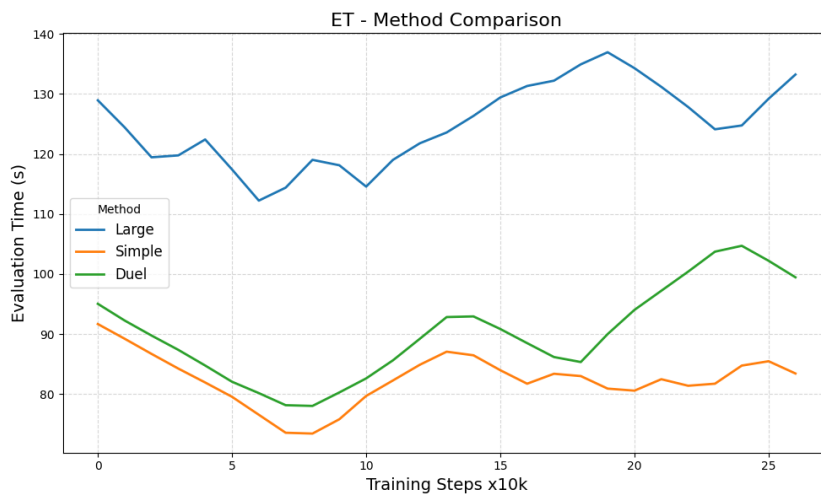
Shared Network: Architecture with a shared network for both policy and value (introduced in Section 5.3.1) estimation achieves consistent win rates, peaking at average of 75%, and relatively stable schedule quality without dramatically dropping. Most importantly, the evaluation time are comparatively the lowest compared to the other two structures and the curve exhibits a downward trend, maintaining an average of 80 seconds (0.80 seconds per board), making this approach efficient for inference.

Separate (duel) Value and Policy Network: This architecture use separate forward neural network for policy and value, achieving the highest schedule quality among the three approaches. At the end of training, separate network structure achieves similar performance to shared network on win rate at 70%. Moreover, this improvement comes at a higher computational cost, increasing the evaluation time to an average of 90 seconds (0.90 seconds per board), reflecting the added complexity of managing two distinct networks.

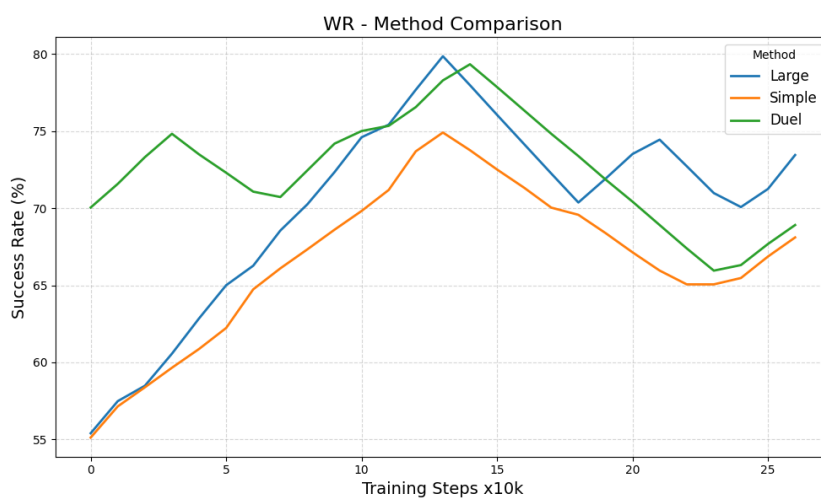
Large Shared Network: The large shared network integrates convolutional layers and 10 residual blocks, each with 640 neurons, following the design principles of AlphaZero [Silver et al., 2017], to enhance feature extraction and alleviate gradient vanishing issues. However, this architecture results in a noticeable decline in schedule quality, suggesting that simply increasing network capacity does not necessarily lead to improved decision-making in the context of traffic control. Additionally, the evaluation time increases significantly, averaging 130 seconds (1.30 seconds per board), which highlights the inefficiency of this architecture and its limited practicality for real-time traffic management applications.

The results underscore a trade-off between performance and computational efficiency across the examined neural network structures. The compact shared network offers a balanced approach with reasonable evaluation times and moderate performance, though it is limited by a lower upper-bound in schedule quality. The separate value and policy network, while achieving higher win rates and schedule quality, exhibits unstable performance and a higher computational cost. On the other hand, the large shared network proves to be the least effective, suffering from declining win rates and policy quality, along with significantly longer evaluation times, making it an impractical choice.

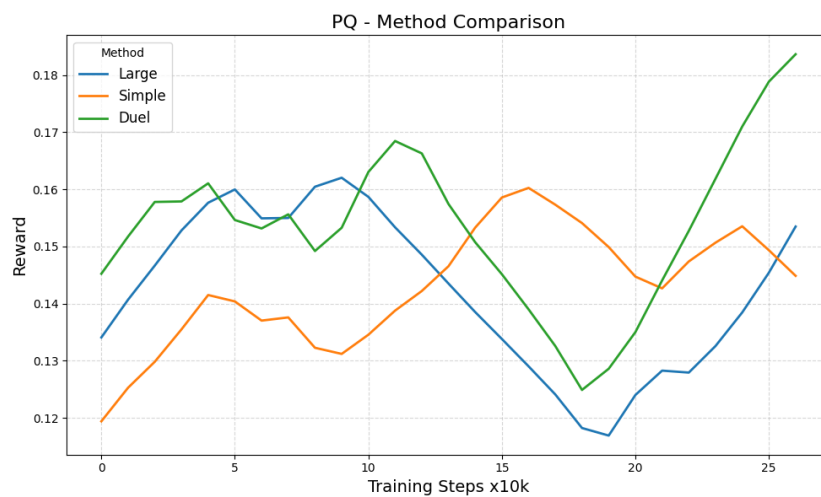
The challenges introduced by scaling to larger networks can be attributed to two main factors. First, increased model complexity may lead to overfitting on local traffic patterns while failing to generalize across diverse traffic conditions. Moreover, deeper architectures can introduce optimization difficulties, such as unstable gradient propagation or suboptimal convergence in large and dynamic environments [Soydaner, 2020]. Second, the limited depth of the current network



(a) Evaluation Time



(b) Win Rate



(c) Policy Quality.

Figure 8.5: Performance comparison between different NN structures.

constrains the effectiveness of residual connections. Residual blocks are known to be most beneficial in wide and sufficiently deep networks, typically with more than 20 linear layers [Wu et al., 2019a].

In such architectures, residual connections help preserve gradient flow and enable efficient backpropagation through earlier layers, mitigating gradient degradation. However, in this case, the network comprises only three repeated convolutional layers, and the overhead introduced by residual blocks outweighs their benefits, contributing minimally to performance gains.

Additionally, empirical results suggest that the board state in this problem is typically represented as a sparse matrix. This sparsity reduces the efficiency of convolutional operations, which are better suited for dense input data, as seen in games like Go or Chess. Consequently, the convolutional layers contribute less to performance in this context compared to traditional board games with denser representations. These factors combined suggest that the large shared network is not an optimal architecture for the task at hand, as its design fails to align with the characteristics of the board representation and the computational requirements of the problem.

Based on these observations, the compact shared network is the most suitable choice as it provides the best trade-off between performance and computational efficiency while maintaining a relatively low evaluation time.

Evaluation of PNMCTS Enhancements on Training Boards

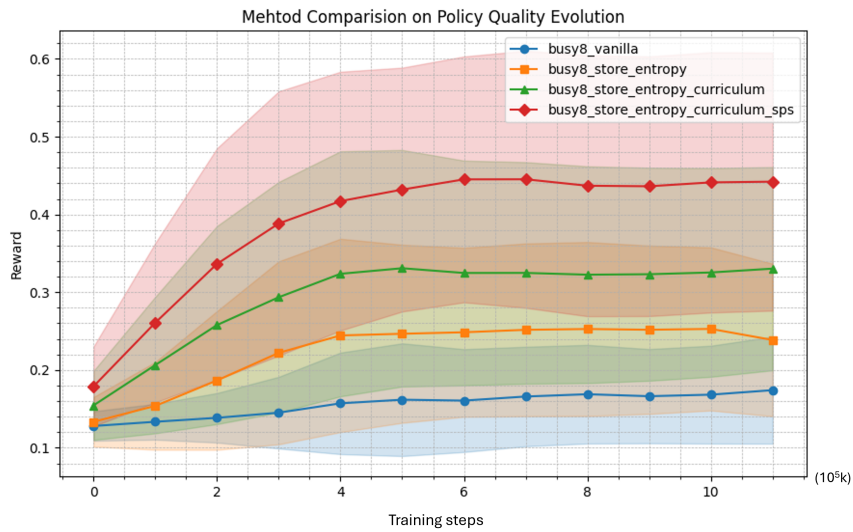


Figure 8.6: Ablation study comparison on schedule quality Evolution. Median rewards and corresponding variances are plotted for different enhancements across training steps.

Figure 8.6 depicts the progression of median rewards over training steps (1.1×10^6) for each enhancement applied on 400 busy boards. As evident, vanilla (standard) NMCTS struggles to improve the solution quality, achieving only marginal reward gains throughout training. In contrast, *busy8_store_entropy* demonstrates a moderate improvement due to enhanced exploration facilitated by the entropy term and best-so-far (BSF) re-sampling, denoted as PNMCTS-RE. Introducing curriculum learning (*busy8_store_entropy_curriculum*) further accelerates training, leading to faster convergence to higher rewards, denoted as PNMCTS-REC. Finally, the addition of SPS (*busy8_store_entropy_curriculum_sps*) yields the best performance, achieving significantly higher and more stable median rewards, denoted as PNMCTS-RECS.

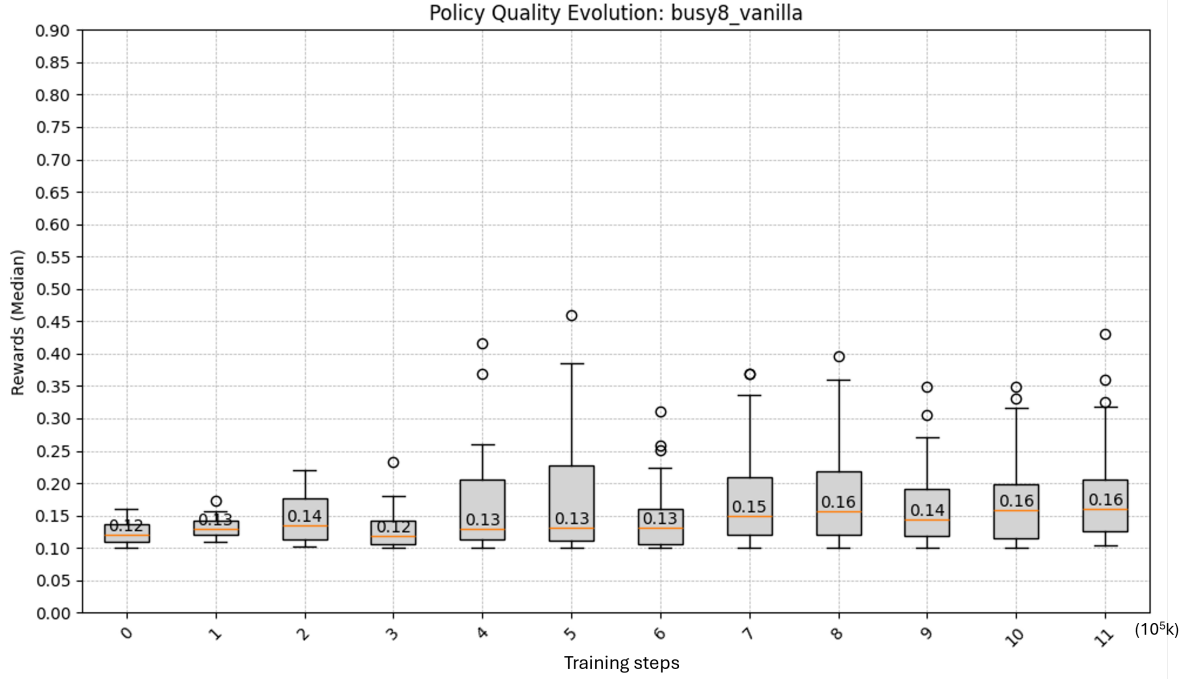


Figure 8.7: Median rewards with standard deviations using standard NMCTS

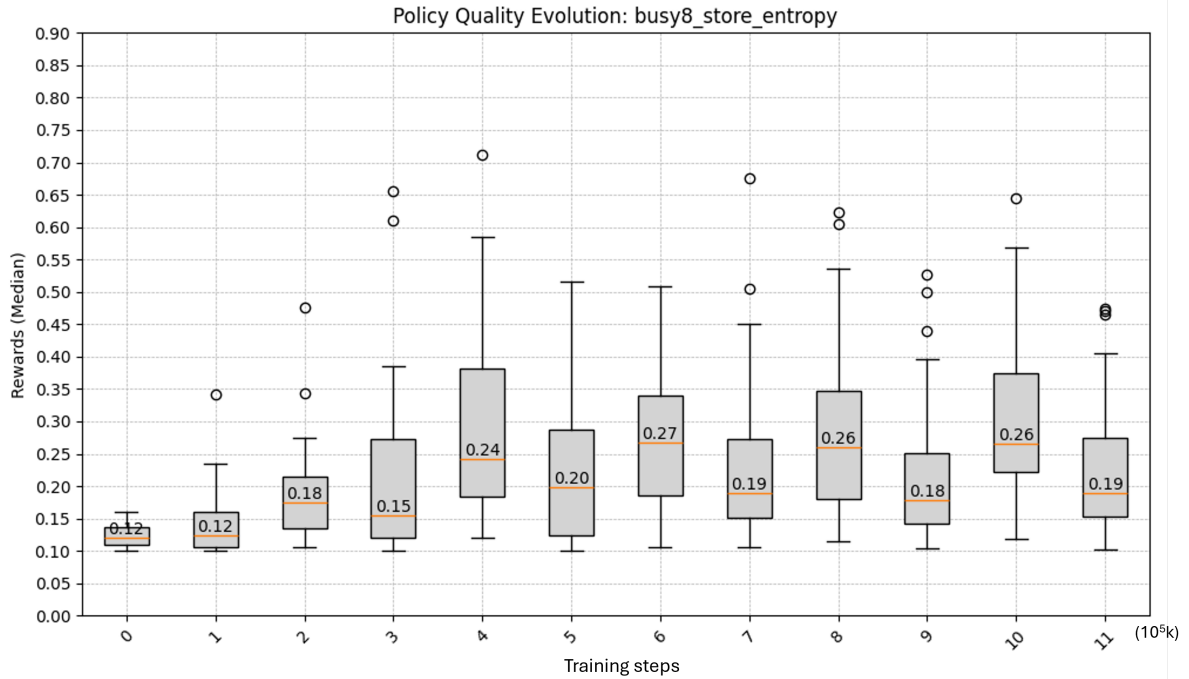


Figure 8.8: Median rewards with standard deviations using PNMCTS-RE

To study the impact of each modification to the standard NMCTS, we examine the step-wise distribution of rewards in the training process using box plots from Figure 8.7 to Figure 8.10. The results from the ablation study reveal the distinct performance improvements brought about by each proposed enhancement to PNMCTS. Starting with the standard NMCTS in Figure 8.7, the model demonstrates limited improvement in median rewards throughout training. The median

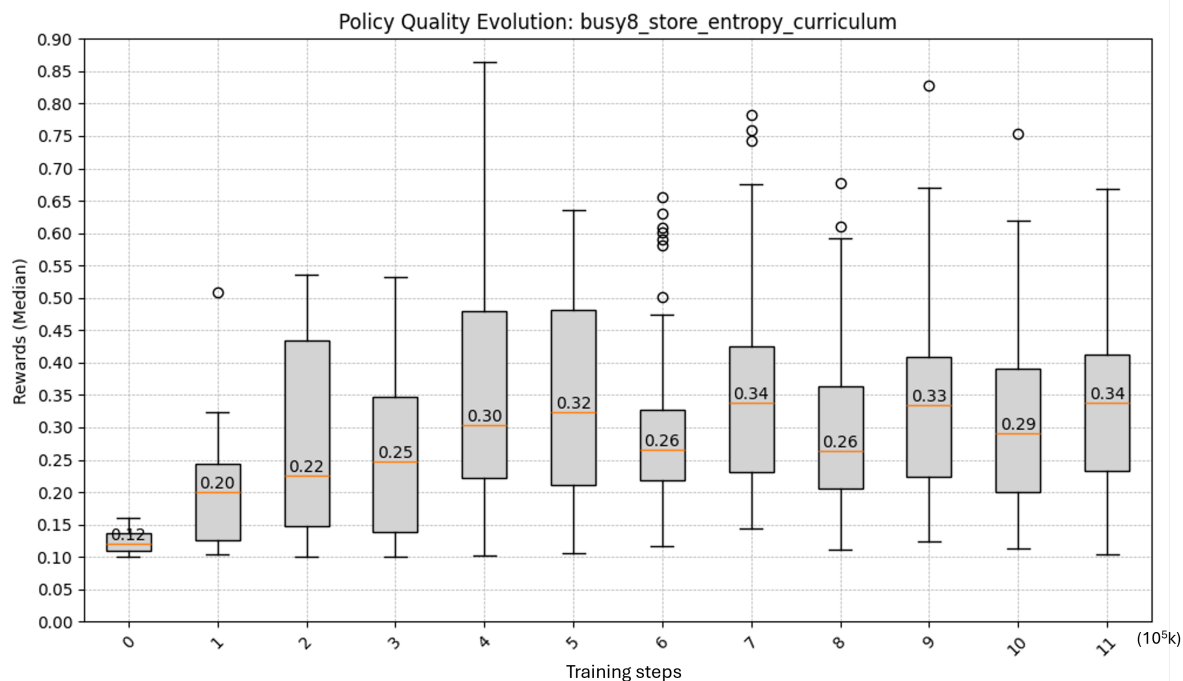


Figure 8.9: Median rewards with standard deviations using PNMCTS-REC

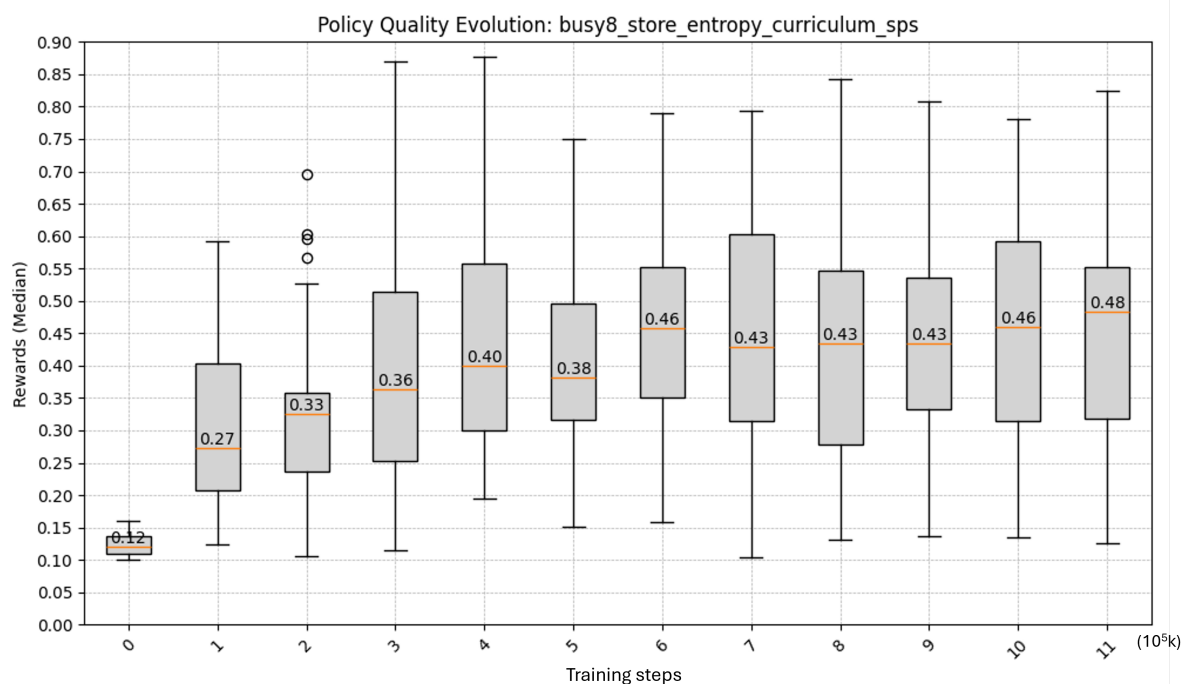


Figure 8.10: Median rewards with standard deviations using PNMCTS-RECS

reward remains consistently low, achieving a final value of 0.16, with minimal variance across training steps. **This indicates that the baseline struggles to explore or adapt to the complexities of the busy board configurations.**

The inclusion of BSF resampling and entropy loss in Figure 8.8 significantly enhances exploration capabilities. The median reward increases notably compared to the baseline, stabilising around

0.26–0.27 by the end of training. Furthermore, the increased variance in reward distribution suggests that the model discovers higher-reward strategies during training. However, the fluctuation in later steps highlight the challenge of maintaining stable performance with BSF resampling and entropy loss.

Adding curriculum learning in Figure 8.9 leads to more structured and stable policy improvement. By introducing simpler boards during early training and gradually transitioning to more complex scenarios, i.e., busy boards, the median reward improves to 0.34 by the end of training. The smoother reward progression and consistent variance suggest that curriculum learning enhances generalisation and allows the model to adapt effectively to increasing board complexities.

Finally, the utilization of SPS in Figure 8.10 achieves the highest and most stable performance among the methods evaluated. The combination of BSF resampling, entropy loss, curriculum learning, and SPS enables the model to robustly handle busy boards while maintaining reasonable inference times. The final median reward of 0.48 reflects a significant improvement over all other methods. **The high variance indicates that the policy can solve a range of boards, from simple to complex, while maintaining reasonable generality with a balanced trade-off between exploration and exploitation.** These findings collectively demonstrate the complementary effects of the proposed modifications in enhancing schedule quality and robustness.

Evaluation of PNMCTS Enhancements on Testing Boards

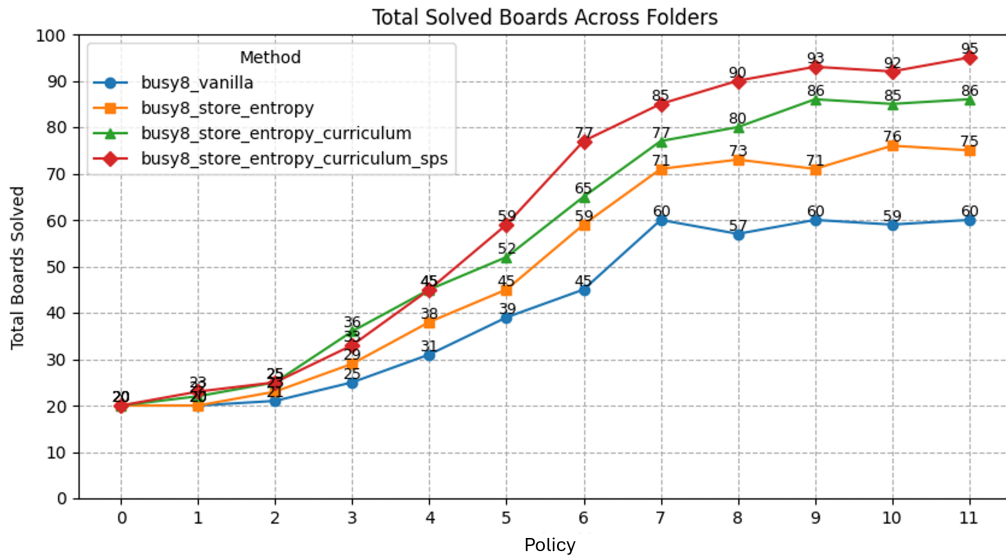


Figure 8.11: Total number of solved board in the evaluation boards.

To evaluate practical performance improvements, we analysed the total number of solved boards in the evaluation dataset, which consists of 100 unseen busy boards. Intermediate policies are stored every 10^5 steps during training, resulting in 11 trained policies, plus the initial policy. The progression of solved boards, shown in Figure 8.11, demonstrates the effectiveness of the proposed approach. Similar trends are observed, with more boards being solved compared to implementations using re-sampling and entropy loss. The introduction of curriculum learning further enhances

performance by providing structured learning pathways. Notably, the addition of SPS consistently maintains high performance in later, more challenging stages, achieving a maximum solution rate of up to 95% at convergence.

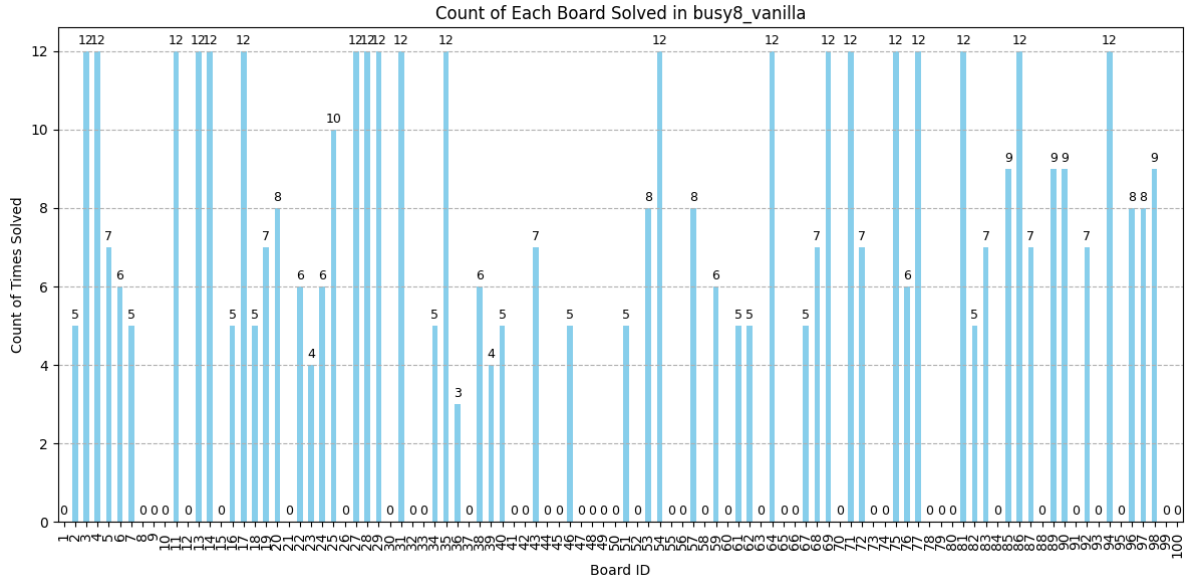


Figure 8.12: Times each board is solved repeatedly using standard NMCTS

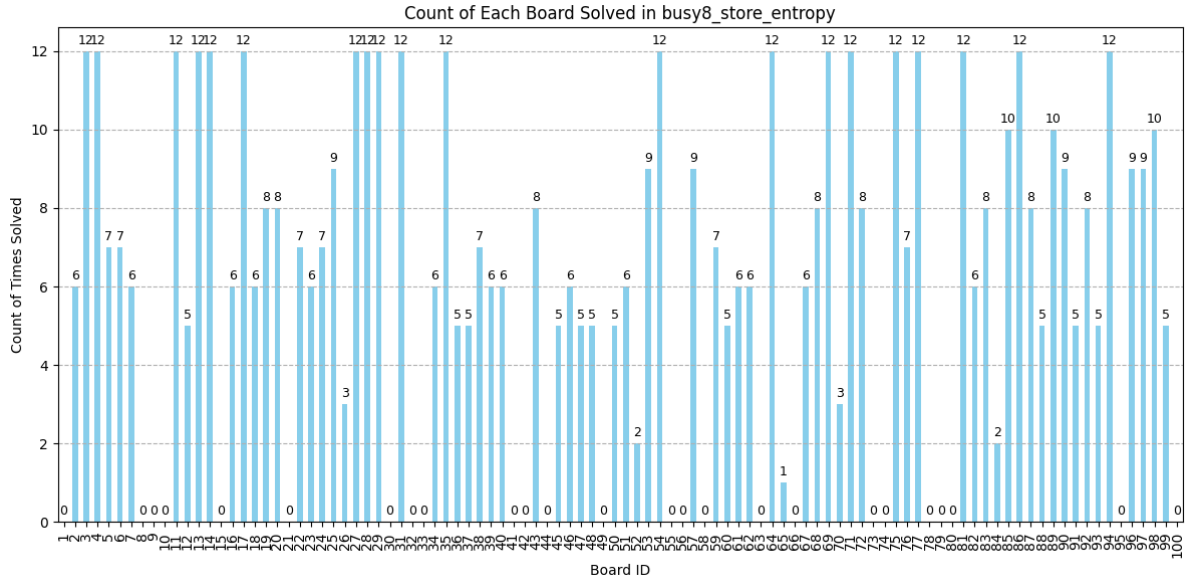


Figure 8.13: Times each board is solved repeatedly using PNMCTS-RE

From Figure 8.12 to Figure 8.15, the performance improvements can be interpreted from two perspectives:

1. **Increase in the total number of solved boards:** The overall rise in successfully solved boards indicates improved generalisation capability, enabling the proposed method to address a broader range of problem instances. This highlights the robustness of the approach under diverse conditions.

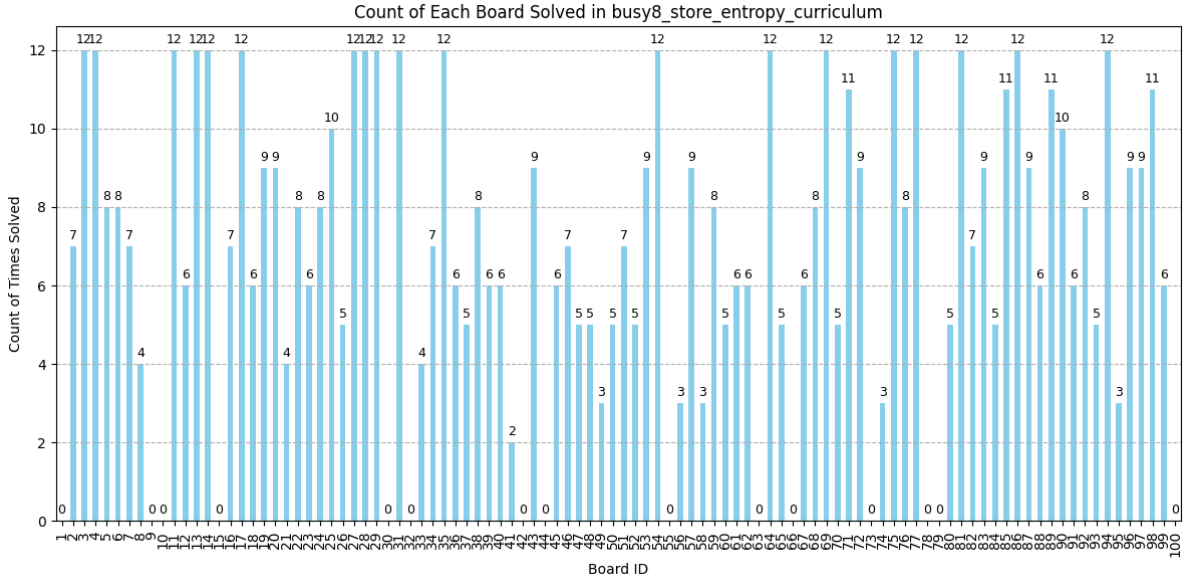


Figure 8.14: Times each board is solved repeatedly using PNMCTS-REC

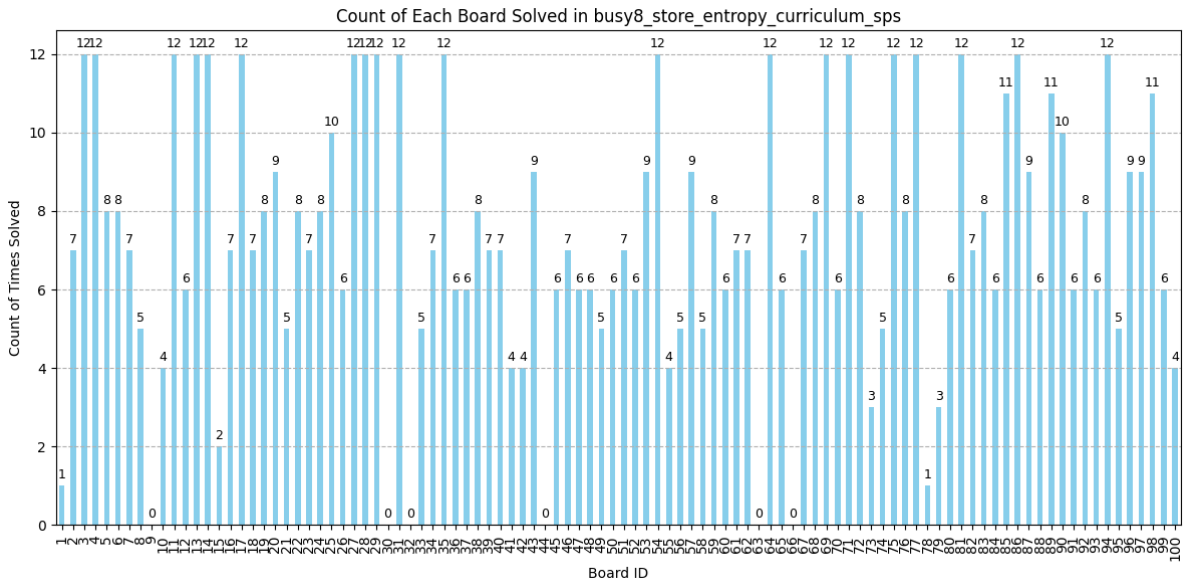


Figure 8.15: Times each board is solved repeatedly using PNMCTS-RECS

2. **Increase in the number of consistently solved boards:** A higher count of boards solved repeatedly suggests that the learned policy effectively retains knowledge from past experiences. This mitigates the risk of catastrophic forgetting and supports stable, continual learning, reinforcing the method's ability to preserve and refine effective strategies over time.

We also observe that in busy intersection scenarios, the standard NMCTS algorithm tends to take larger steps to navigate the decision space, as shown in Figure 8.16. While this approach speeds up decision making, it often sacrifices solution quality, leading to suboptimal results. In contrast, the PNMCTS policy executes finer-grained steps, enabling more precise decision-making. By making smaller adjustments, PNMCTS explores solutions that standard NMCTS may overlook, allowing it to adapt more effectively to complex intersection scenarios where subtle refinements are crucial for

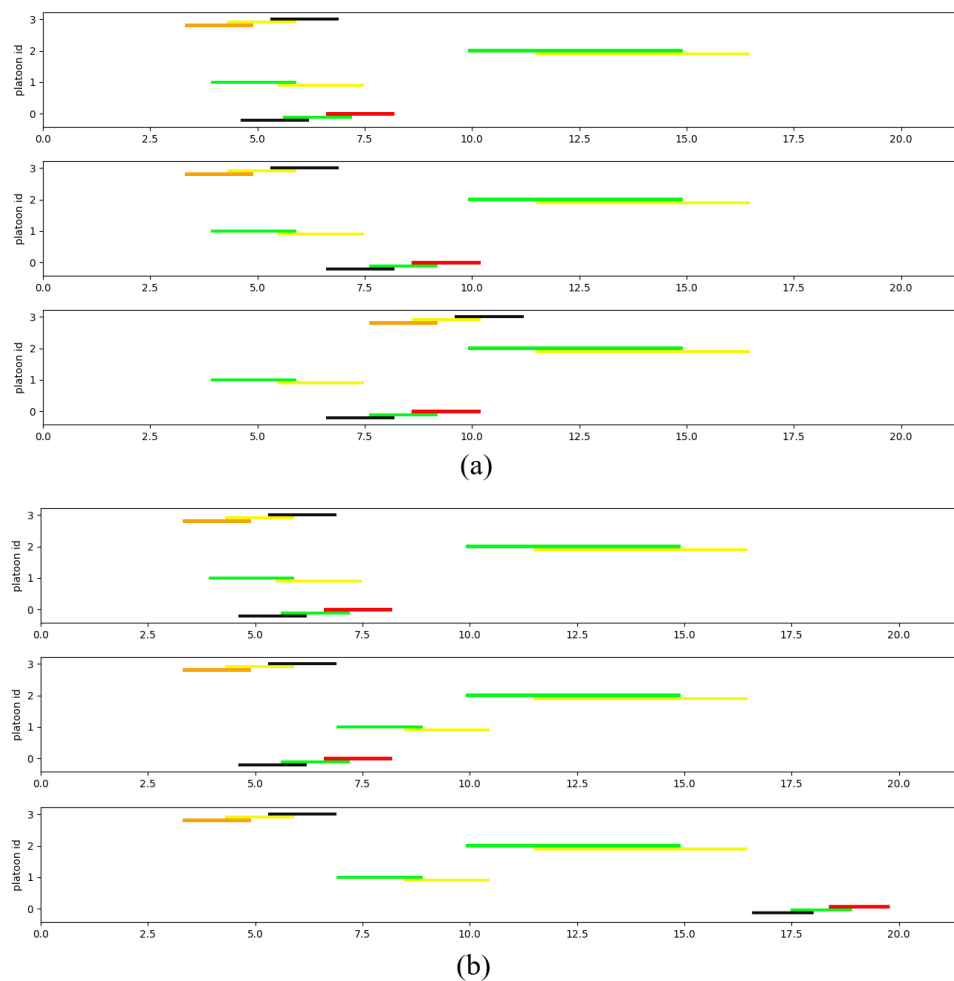


Figure 8.16: Comparison between optimal and sub-optimal schedules. (a) An optimal solution use 0.9 second.(b) A sub-optimal solution use 0.2 second. Note that the Last action is explicitly different as the PNMCTS policy performs a finer-grained action leading to a better solution.

optimal performance.

However, the adoption of smaller, finer-grained steps comes at the expense of computational efficiency. The increased granularity results in a longer decision-making duration because the trained policy requires multiple incremental steps to converge on an optimal solution. This phenomenon becomes more evident in scenarios where resolving the intersection involves a series of decisions, making the process more time consuming, as shown in Figure 8.17.

Table 8.2: Success rate comparison at synthetic boards

Clear intersection				
Policy	Encountered Boards	Success Rate	Unseen Boards	Success Rate
Standard NMCTS	265/400	66.25%	55/100	55.0%
PNMCTS-RECS	385/400	96.25%	85/100	85.0%
Busy intersection				
Policy	Encountered Boards	Success Rate	Unseen Boards	Success Rate
Standard NMCTS	120/400	30.0%	20/100	20.0%
PNMCTS-RECS	360/400	90.0%	95/100	95%

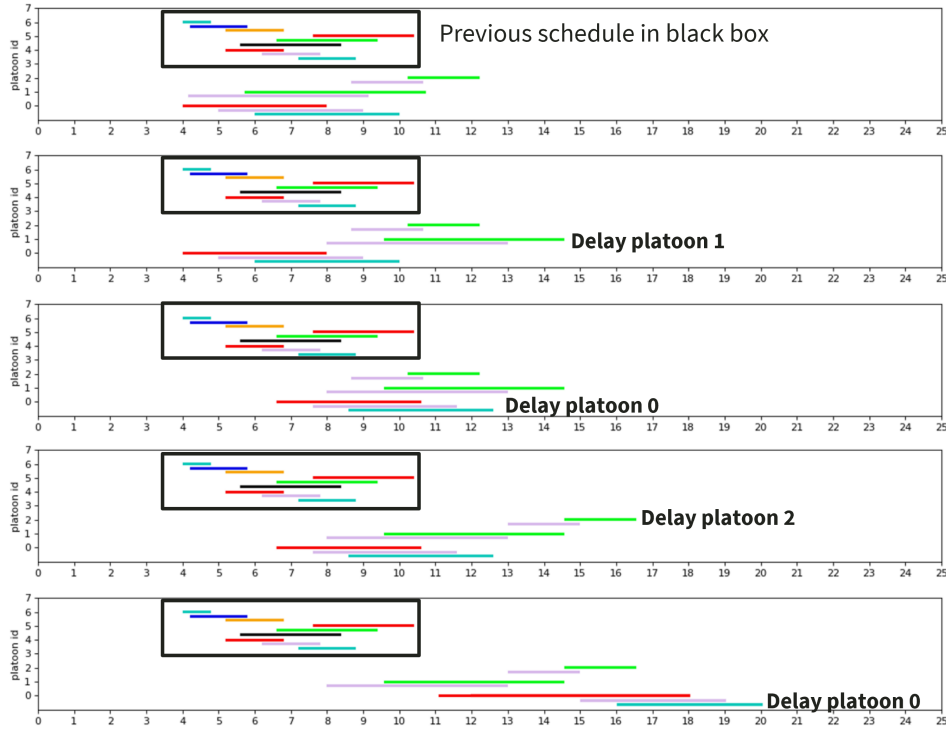


Figure 8.17: An optimal crossing schedule provided by a PNMCTS agent at a busy intersection from top to bottom, platoon delay sequence: initial board $\rightarrow$ platoon1 $\rightarrow$ platoon0 $\rightarrow$ platoon2 $\rightarrow$ platoon0. The total time usage is 1.2 seconds which is longest among all evaluation boards.

A straight comparison of the success rates between standard NMCTS and PNMCTS-RECS is presented in Table 8.2. The PNMCTS policy outperforms standard NMCTS in both clear and busy intersections. Notably, in clear intersection cases, PNMCTS-REC makes decisions solely based on the policy network, as these simpler boards are easier to learn and solve. Given sufficient policy network capacity, the PNMCTS-RECS policy successfully solved 96.25% of training boards and 85% of unseen boards.

Despite these improvements, there remain five boards that are never solved within the given time constraints. This limitation indicates potential areas for further optimization, such as enhancing the exploration-exploitation strategy, introducing more effective learning mechanisms, or adapting the algorithm to better handle particularly challenging boards. By integrating these enhancements, the system could achieve higher problem-solving efficiency, reduce the number of unsolved boards, and further improve its capability to generalize across various scenarios.

Evaluation of PNMCTS on Multi-intersection Grid

This study investigates the performance of the proposed PNMCTS agents as they are progressively deployed at intersections, replacing fixed-time controllers. The evaluation focuses on two key metrics: **ATT** and **TT**. By analysing these metrics, this study assesses the impact of agent deployment rates on control strategies for improving traffic efficiency, reducing congestion, and enhancing overall network performance.

The experiments were conducted on a homogeneous grid network where each edge has a length of

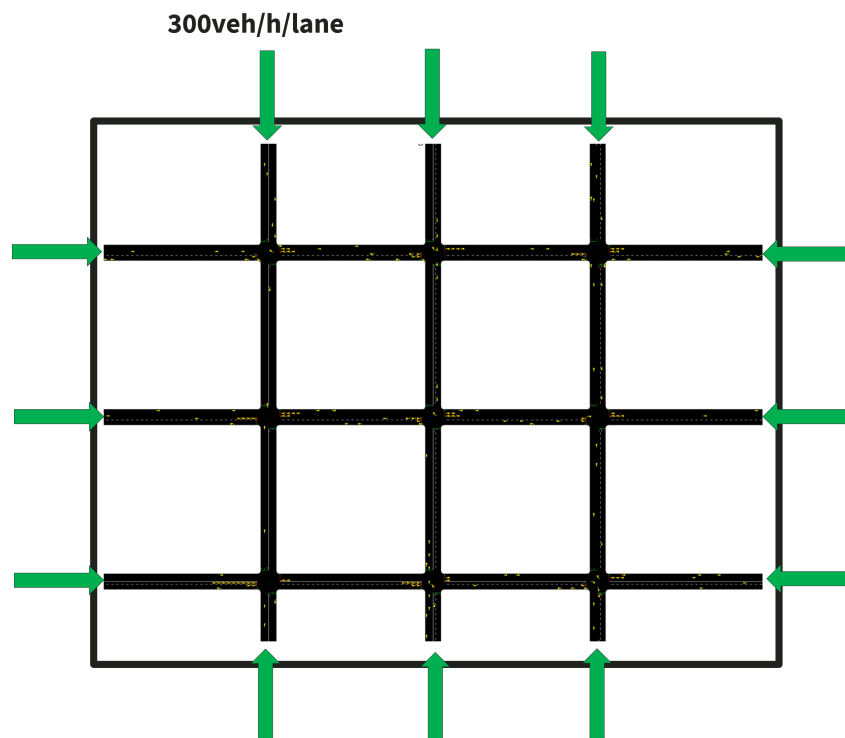


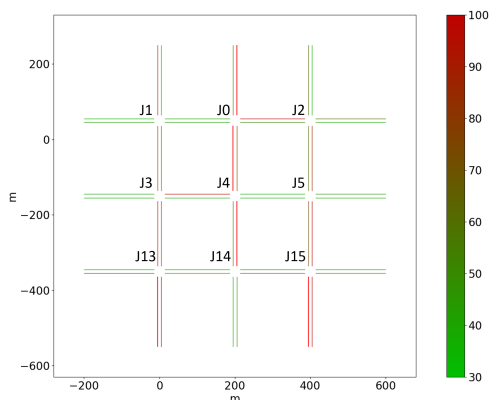
Figure 8.18: Traffic flows are evenly inserted from the marginal road to simulate a correct traffic patterns.

200 meters, and vehicles travel at a constant speed of 5 m/s. Traffic starts and ends at the marginal edges to simulate continuous vehicle movement across the network. A uniform traffic demand of 300 veh/h/lane is applied, as depicted in Figure 8.18.

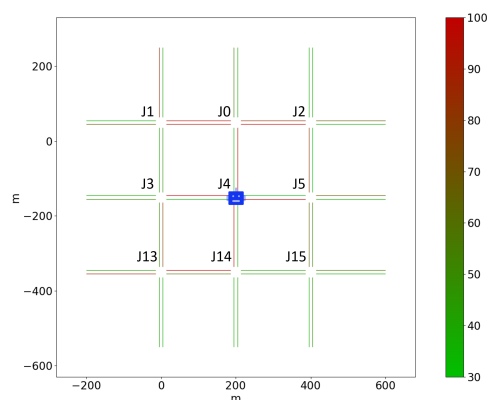
In SUMO simulations, vehicles sometimes execute lane changes too late, leading to lane-changing conflicts near intersections when lateral vehicles aim for each other's lanes. This behaviour can cause lane change blockages, disrupting traffic flow. To mitigate this issue, vehicles are required to complete lane changes at the beginning of each edge. If a vehicle fails to change lanes to its **best** lane within the first 5 seconds of its entry point, it is removed from the network to prevent congestion and maintain overall traffic efficiency. Agent-controlled junctions in the grid are equipped with identical PNMCTS solvers that operate independently to schedule intersection crossing.

Initially, the deployment of a single agent-controlled intersection in the network's centre (as shown in Figure 8.19(b)) leads to an increase in ATT in the system as shown in Figure 8.21(a). This increase can be attributed to congestion build-up in its downstream lanes. The presence of an autonomously controlled intersection creates a bottleneck effect, as the traditional traffic control mechanisms surrounding it are unable to efficiently manage the altered traffic patterns, as depicted in Figure 8.20.

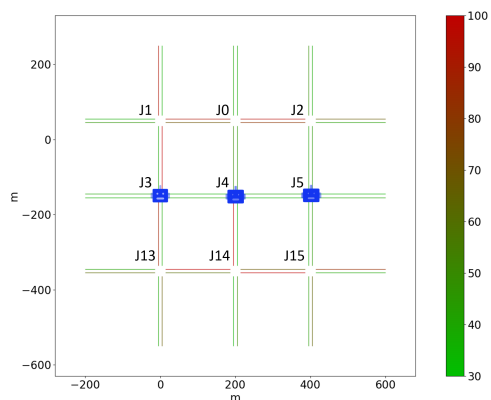
The gradual expansion of agent-based control to additional intersections leads to a noticeable reduction in network congestion as shown in Figure 8.19. The greener shading in setup 6 highlights lower ATT, where fully agent-controlled intersections result in near free-flow traffic conditions. These findings indicate that the intelligent control system effectively reduces queue lengths at intersections, directly improving overall traffic flow and travel time.



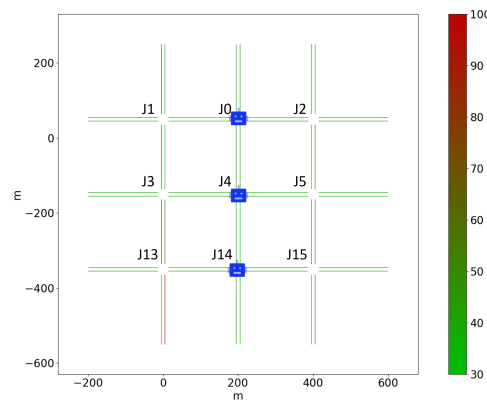
(a) Scenario 1: All intersections controlled by fixed-time traffic lights.



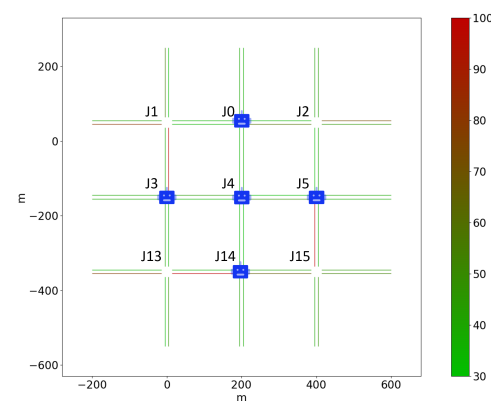
(b) Scenario 2: One intersection controlled by an agent-based controller.



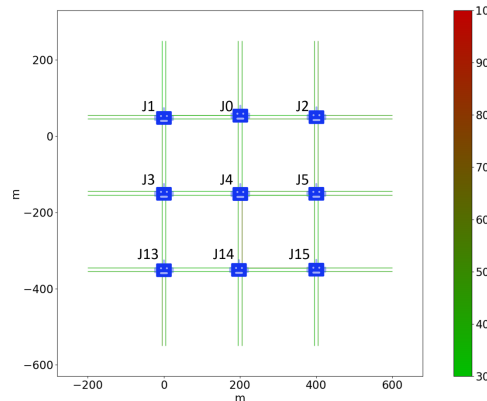
(c) Scenario 3: Three intersections horizontally controlled by agent-based controllers.



(d) Scenario 4: Three intersections vertically controlled by agent-based controllers.



(e) Scenario 5: Five intersections controlled by agent-based controllers.



(f) Scenario 6: All intersections controlled by agent-based controllers.

Figure 8.19: Progressive replacement of fixed-time traffic light controllers with agent-based controllers across the network. Controlled intersections reduce queue build-up in incoming lanes, as evidenced by reduced travel times (highlighted in green). A fully agent-controlled network can achieve near free-flow travel conditions.

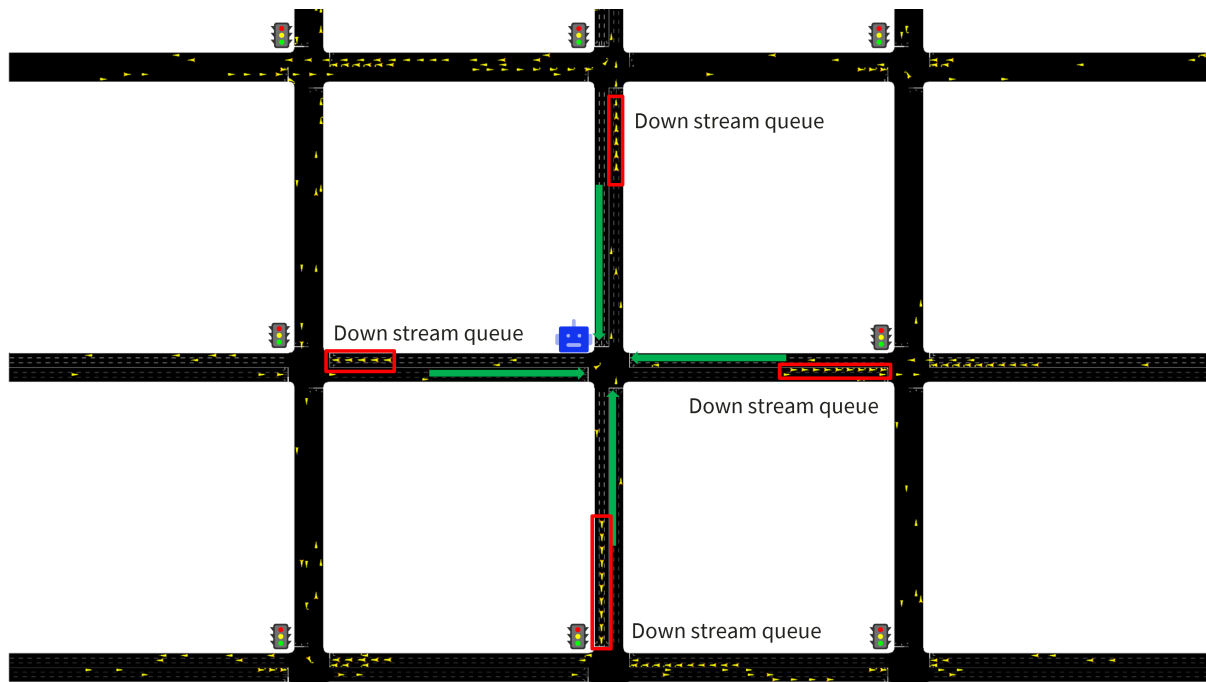


Figure 8.20: Single agent deployment leads to more downstream congestion.

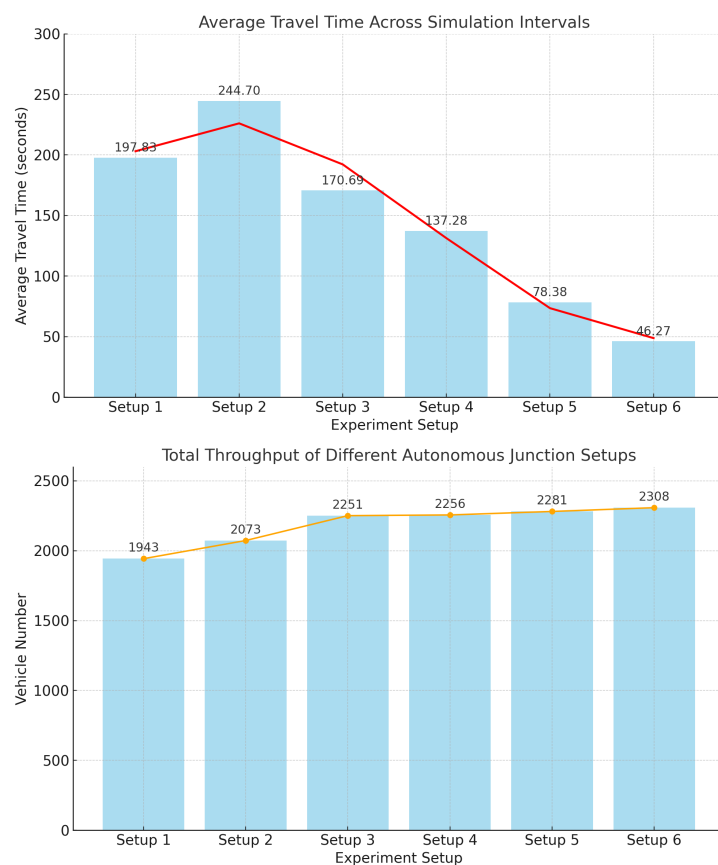


Figure 8.21: (a): Average travel time and (b): total throughput of the network for scenarios 1-6.

We observe a continuous reduction in travel time and improvement in total traffic throughput, as shown in Figure 8.21(a) and Figure 8.21(b), respectively. This trend can be explained by the

dynamics of traffic flow and the efficiency of agent-based intersection control. As more intersections are managed by intelligent agents, traffic becomes better coordinated, reducing delays caused by bottlenecks and minimizing stop-and-go driving conditions. This improved coordination allows vehicles to move through the network smoothly, lowering the average travel time. However, network throughput, defined as the total number of vehicles passing through the network in a given period, eventually reaches a saturation point. This plateau occurs because the network's maximum capacity is limited by its physical characteristics, such as the number of lanes, speed limits, and vehicle dynamics. While intelligent agents optimize the movement of vehicles already within the network, they cannot increase the total number of vehicles the network can handle. As a result, although vehicles travel more efficiently and quickly, the overall carrying capacity of the network remains unchanged.

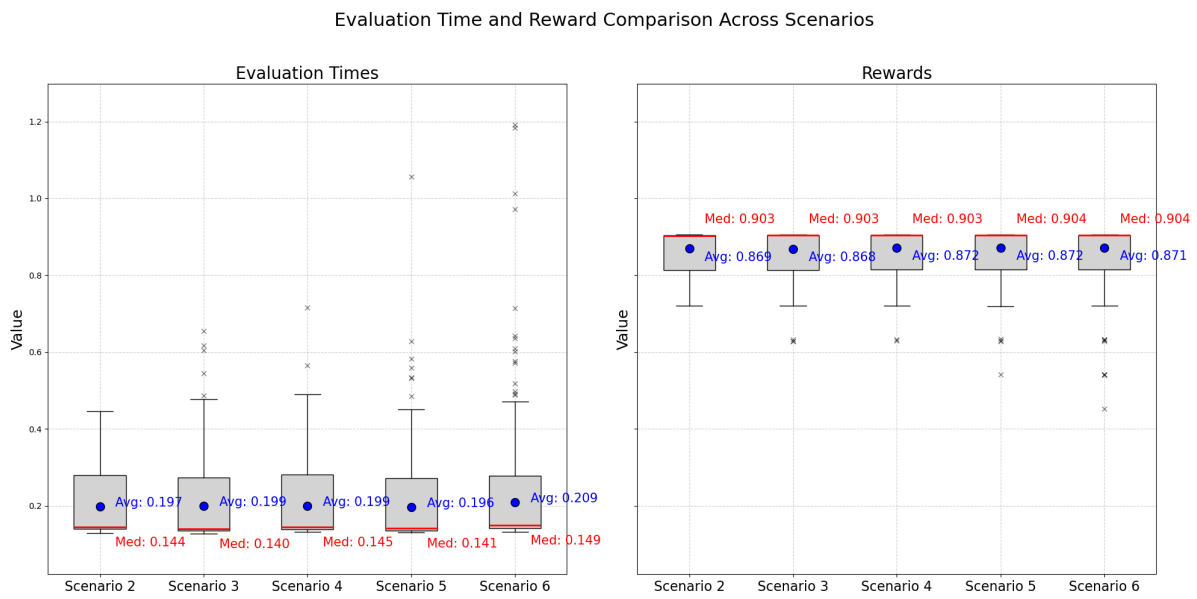


Figure 8.22: Progressive replacement of fixed-time traffic light controllers with agent-based controllers across the network. Schedule reward and inference time details.

We analyse the effect of progressively deploying agent-based traffic light controllers on both solution quality (reward) and evaluation time (inference latency), as shown in Figure 8.22. The analysis spans five deployment scenarios, from **Scenario 2** (one intersection controlled by an agent) to **Scenario 6** (all six intersections agent-controlled).

The right plot of Figure 8.22 shows the distribution of solution quality across scenarios. The median reward remains stable, around **0.903–0.904**, indicating that deploying more agents does not compromise policy performance. The mean reward improves slightly, increasing from **0.869** in Scenario 2 to **0.871** in Scenario 6. However, the number of outliers increases as agent deployment expands, suggesting that while the average performance improves, the likelihood of occasional suboptimal behaviour also grows under fully distributed coordination. The interquartile range remains similar across all scenarios, showing no clear trend in variability among the central data.

The left plot presents evaluation time per decision. Here, the median evaluation time grows slightly from **0.140s** in Scenario 2 to **0.149s** in Scenario 6, while the mean increases from **0.197s** to

0.209s. The IQR remains relatively tight across scenarios, though higher agent counts result in more frequent and extreme outliers, indicating occasionally latency spikes caused by increased traffic flow and computation overhead. Still, the overall inference latency remains low, demonstrating the scalability of the system.

Together, these results show that expanding agent deployment enhances average solution quality with a modest increase in evaluation time, while introducing a higher chance of rare suboptimal or slower responses.

8.2.4 Summary of Intersection Scheduling

This section presents a comprehensive evaluation of the proposed PNMCTS method across diverse traffic conditions and network topologies, including single intersections, multi-intersection grids, and progressive agent deployment scenarios. Through extensive experiments and ablation studies, PNMCTS consistently demonstrates superior performance over baseline methods—such as FIFO, fixed-time control, actuated control, and state-of-the-art methods like PressLight—in terms of travel time reduction, throughput improvement, and schedule quality.

The results underscore the importance of strategically deploying agent-based intersection controllers to enhance overall network efficiency. By enabling decentralized yet coordinated scheduling decisions, PNMCTS effectively reduces crossing delays and inference time. Moreover, it serves as a robust foundation for providing fast, short-term intersection scheduling feedback, which supports higher-level traffic network optimization objectives.

However, the current implementation uses a fixed platoon length of up to five vehicles during both training and evaluation. This rigid rule—grouping vehicles into a platoon if fewer than five, or splitting them if more—proves effective under low-density traffic but becomes inefficient in high-density scenarios. To address this limitation, we propose a dynamic platoon selection policy that functions as a higher-level decision-making mechanism, enabling adaptive vehicle grouping based on real-time traffic conditions. The evaluation of this enhancement is presented in the following section.

8.3 Evaluation of Traffic Network Optimization

This section evaluates the proposed HRL-MAPPO framework in the context of system-wide traffic control across various network configurations. Building on the promising results from intersection-level scheduling, we investigate the framework’s scalability and its capacity to coordinate decentralized decision-making for efficient traffic optimization in multi-intersection environments. The evaluation includes performance comparisons against both traditional and state-of-the-art learning-based baselines on single intersections and homogeneous 3×3 grid networks. Furthermore, we extend the experiments to large-scale, heterogeneous real-world deployments—specifically, the Dublin city network—under both zero-shot and fine-tuning scenarios. In addition, we analyse critical design choices within the hierarchical architecture, including reward sharing mechanisms and dynamic platoon selection strategies, to demonstrate how high-level

decision-making contributes to improved global throughput, reduced average travel time, and enhanced travel time reliability.

8.3.1 Evaluation of HRL Performance on Single intersection

In this section, we conduct experiments to evaluate the performance of the proposed HRL framework by deploying an isolated PPO agent on a single intersection under various traffic flow patterns. Specifically, we consider four types of traffic conditions: low, moderate, and high constant flow rates (Table 8.3), as well as two dynamic patterns—cyclic square wave (Figure 8.23) and sinusoidal wave (Figure 8.24). These scenarios are designed to assess the agent’s adaptability and robustness across both stationary and time-varying traffic demands.

Table 8.3: Low, moderate, and high traffic flow rates (in vehicles per hour) for a single unsignalized intersection.

Direction	Low	Moderate	High
North-South	100	500	1000
North-East	200	400	800
North-West	50	300	600
South-North	100	450	900
South-East	100	600	1200
South-West	200	300	600
East-North	200	200	400
East-South	100	400	800
East-West	100	400	800
West-North	100	500	1000
West-South	50	200	400
West-East	100	300	600

Table 8.4: ATT Comparison of State-of-the-Art Baselines for Single Unsignalized Methods. Lower the better, and the best results are in bold.

Methods	Low flow rate	High flow rate	Moderate flow rate	Cyclic square	Sinusoidal-type
R-based AIM	124.79	145.28	134.90	139.17	136.22
R&B-based AIM	75.67	120.57	106.91	114.39	112.34
O&B-based AIM	69.79	91.48	86.34	98.68	95.79
DCL-AIM	40.44	89.03	84.26	90.87	72.01
INTEL-PLT	58.31	79.27	73.85	77.10	71.77
COOR-PLT	33.01	38.82	37.52	37.77	37.51
Greedy	28.91	39.25	34.35	31.64	44.10
ISO-PPO	28.34	37.25	33.07	28.64	42.00

Here, we provide a detailed comparison of the proposed method against established baselines.

Table 8.4 presents a comparative analysis of various state-of-the-art traffic management methods under different traffic flow scenarios.

- **Low Flow Rate:** Our proposed method achieves an average travel time (ATT) of 18.34 seconds, which is significantly lower than the nearest competitor, COOR-PLT, with 33.01 seconds. This demonstrates our method’s superior efficiency in managing low traffic conditions by reducing waiting times.

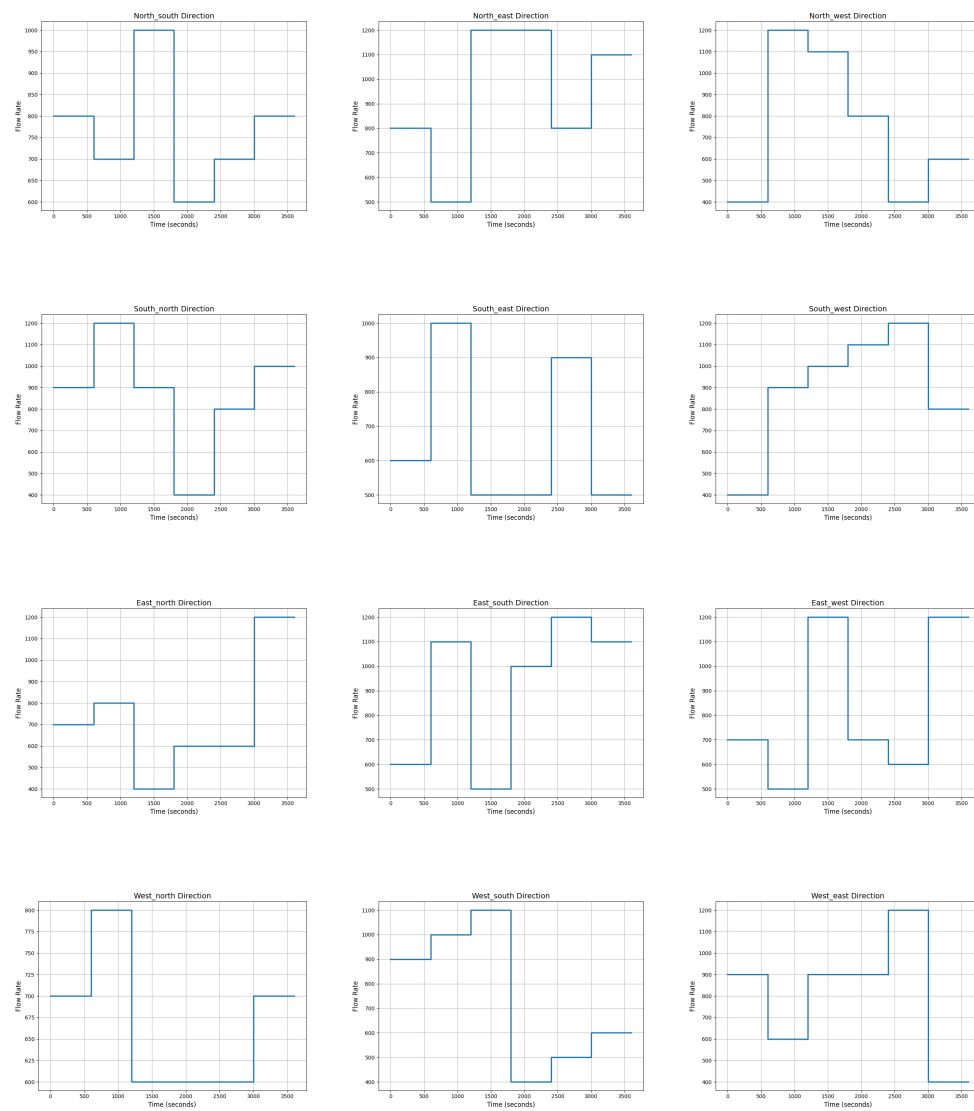


Figure 8.23: Cyclic square functions for traffic flow in various directions.

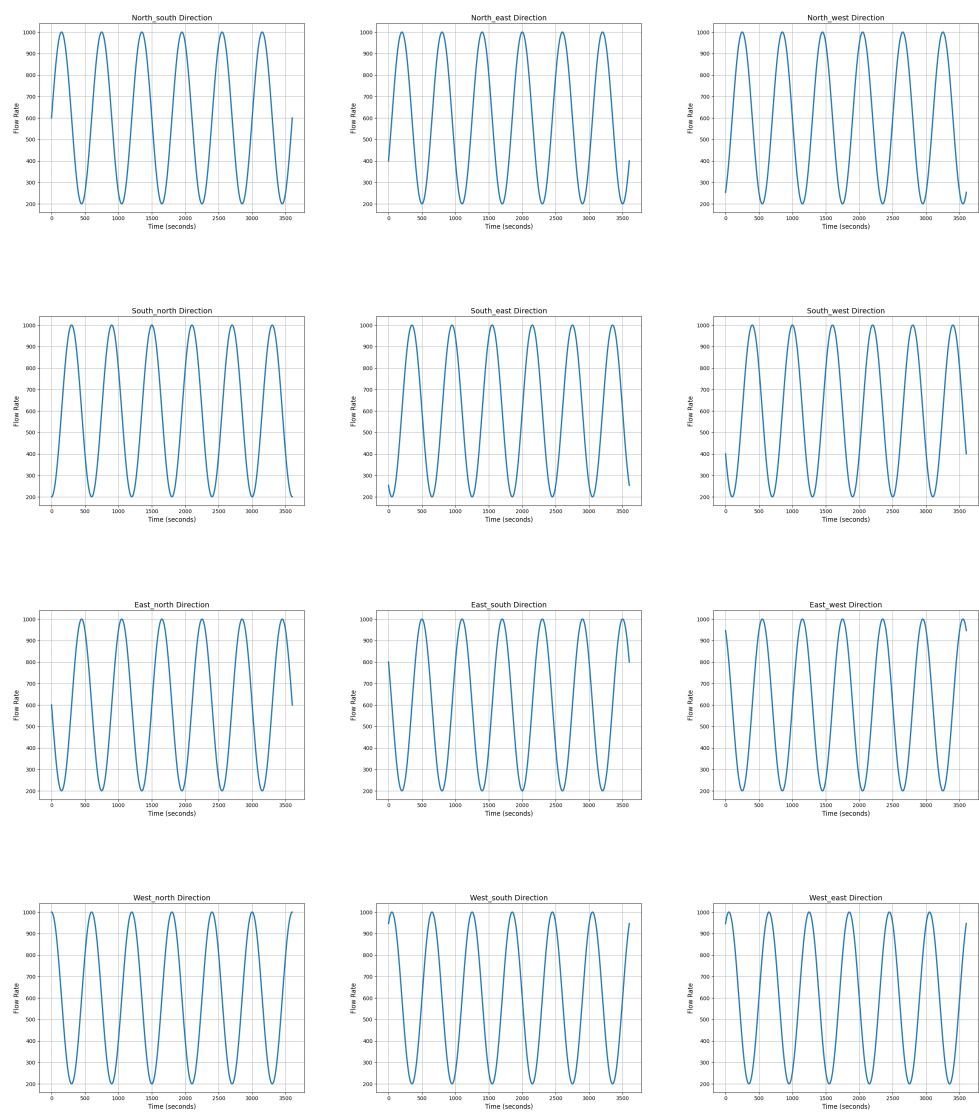


Figure 8.24: Sine wave functions for traffic flow in various directions.

- **High Flow Rate:** In high traffic conditions, our proposed method records an ATT of 37.25 seconds, compared to the next best method, COOR-PLT, which records 38.82 seconds. This slight improvement indicates robustness in handling congested traffic scenarios effectively.
- **Moderate Flow Rate:** For moderate traffic conditions, our method shows an ATT of 30.07 seconds, outperforming COOR-PLT's 37.52 seconds.
- **Cyclic Square Traffic Pattern:** In cyclic square conditions, our method attains an ATT of 28.64 seconds, which is substantially better than the next best 37.77 seconds by COOR-PLT, showcasing our method's adaptability to periodic traffic volume.
- **Sinusoidal-Type Traffic Pattern:** Despite a higher ATT of 42.00 seconds in sinusoidal-type patterns compared to our performance in other categories, our method still outperforms the SOTA traffic light control methods.

8.3.2 Evaluation of HRL-MAPPO Performance on 3×3 Grid Network

We adopt the best-performing configuration identified in previous experiments by setting the maximum platoon length to $L = 5$, as established in Section 8.3.4. Figure 8.25a, Figure 8.25b, and Figure 8.25c present the performance results in terms of TT, ATT, and TTR, respectively. These figures compare the effectiveness of different traffic control strategies under varying traffic demand levels, ranging from 700 to 1000 veh/h/road.

The proposed **HRL-MAPPO** consistently outperforms all baselines across all metrics. It achieves the highest total throughput (up to 5943 vehicles), the lowest average travel time (as low as 117.26 seconds), and the most stable travel time reliability (at 60.8 seconds). These results demonstrate the effectiveness of high-level coordination enabled by the shared policy and joint optimization.

ISO-PPO, while also learning through hierarchical reinforcement learning, performs slightly worse than MAPPO, particularly at higher traffic loads. This performance gap is primarily due to the lack of coordination across intersections, leading to suboptimal global traffic patterns despite good local policies.

Traditional baselines such as **fixed-time** and **actuated** control exhibit limited adaptability, leading to significantly higher average travel times and poor reliability under heavy traffic conditions.

PressLight, a strong RL-based baseline, demonstrates moderate performance but still falls behind both MAPPO and ISO-PPO, particularly in terms of throughput and travel time reliability. This is primarily due to the inherent limitations of traffic light control in terms of crossing efficiency and intersection space utilization. These results underscore the advantages of the proposed hierarchical structure and high-level coordination policy, which enable more effective scheduling and continuous vehicle movement through intersections.

8.3.3 Evaluation of HRL-MAPPO Performance on Dublin City Network

We evaluate the effectiveness of the proposed framework in real-world urban traffic scenarios, focusing on a large-scale, heterogeneous city network. The primary objective is to assess the capability of the proposed HRL-MAPPO approach to enhance traffic efficiency under complex and

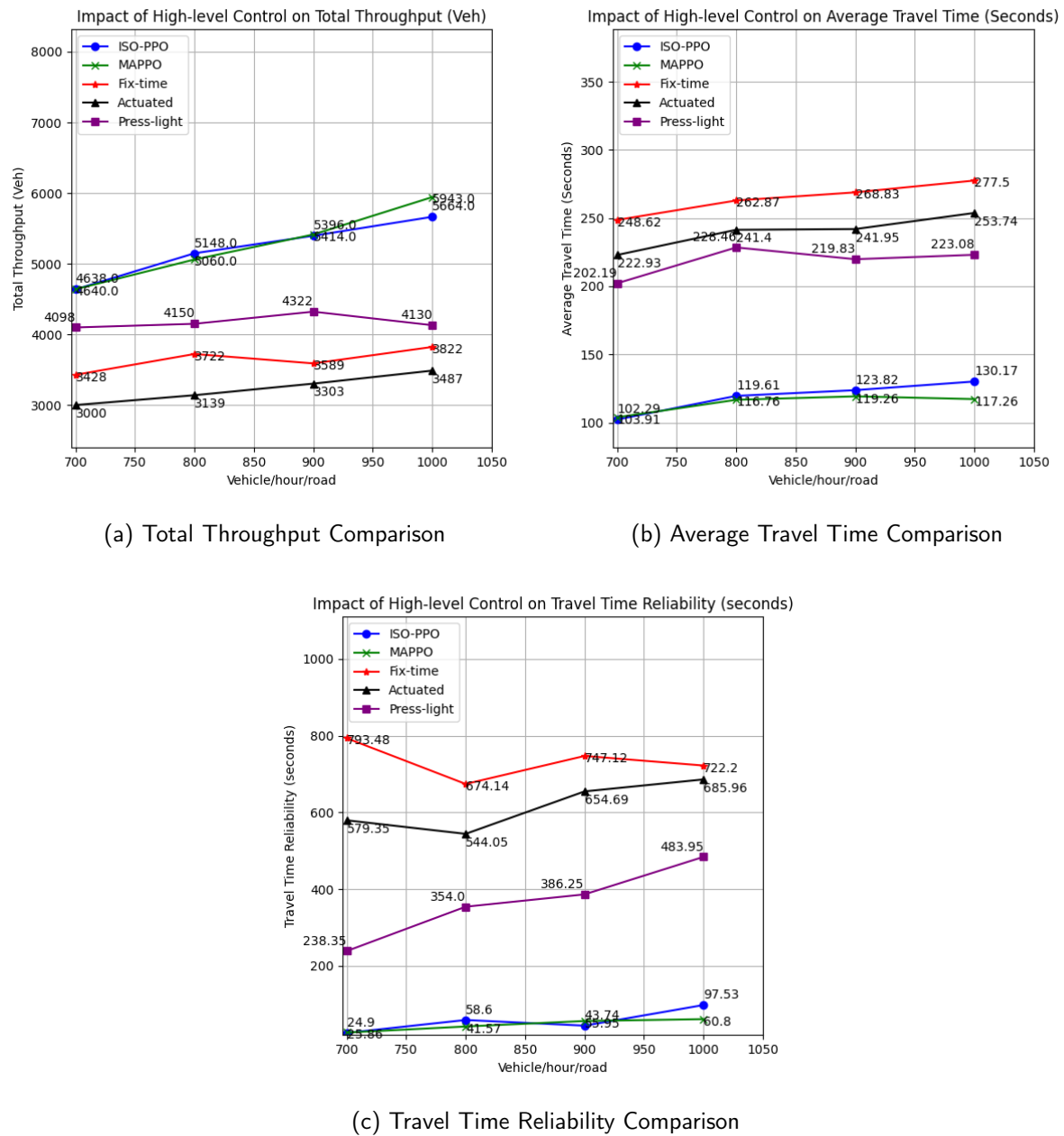


Figure 8.25: Comparison of total throughput, average travel time, and travel time reliability. Higher extra travel time indicates lower travel reliability.

high-demand conditions. The evaluation considers multiple performance metrics, including Travel Throughput (TT), Average Travel Time (ATT), and Travel Time Reliability (TTR), as well as Total Travel Distance (TTD), Waiting Time (WT), and Waiting Count (WC). These additional indicators offer a more detailed characterization of traffic congestion and delay. This comprehensive evaluation not only demonstrates the practical applicability of the proposed method but also underscores its potential to significantly improve urban mobility at scale.

The experiments are conducted using a detailed simulation of Dublin's city network, based on the dataset introduced by [Gu eriau and Dusparic, 2020]. This dataset models traffic patterns over a full 24-hour period and includes high-resolution traffic demand profiles, detailed road topology, and signal configurations derived from empirical observations of Dublin's transportation infrastructure. For evaluation, we focus on the morning peak period (07:00–09:00), simulating 7200 seconds of traffic flow across 1645 heterogeneous intersections with diverse geometries and lane configurations.

Simulation Configuration

We simulate a fully automated traffic environment assuming a 100% penetration rate of connected and autonomous vehicles (CAVs), with no human-driven vehicles involved. Each CAV follows a predefined car-following model incorporating realistic vehicle dynamics and safety constraints. To ensure uniformity, all vehicles are modeled as saloon-type, and a consistent set of parameters is used across all experiments. The baseline traffic light control (TLC) logic used for comparison is implemented using the actuated control strategy described in [Gu eriau and Dusparic, 2020]. Key simulation parameters are summarized in Table 8.5.

Table 8.5: Key simulation parameters for the Dublin city environment.

Parameter	Value
CAV Penetration Rate	100%
Vehicle Type	Passenger car (Saloon)
Car-following Model	IDM
Acceleration Rate	3.0 m/s ²
Deceleration Rate	4.5 m/s ²
Emergency Deceleration	9.0 m/s ²
Maximum Speed Deviation	�5%
Minimum Gap	1.0 m
Time Headway	0.5 s
Driver Imperfection	0.05
Lane-change Strategicness	0.5
Traffic Control Baseline	Actuated

To ensure consistency and comparability across methods, performance metrics are extracted from a representative evaluation window between 31,300 and 31,400 seconds of simulation time. This window corresponds to a steady-state traffic condition during the peak hour, enabling a fair comparison among TLC, zero-shot, and fine-tuning control strategies.

The evaluation comprises two main components: 1) the analysis of the proposed automated collision area detection method, which employs a topology-based scheme coupled with DBSCAN

clustering to identify high-risk interaction zones within intersections, and 2) the assessment of zero-shot and fine-tuning deployment strategies in large-scale traffic control. These experiments demonstrate the framework's effectiveness in mitigating congestion, improving travel reliability, and increasing overall throughput across the city network.

Results of Collision Areas Detection in Dublin City Center

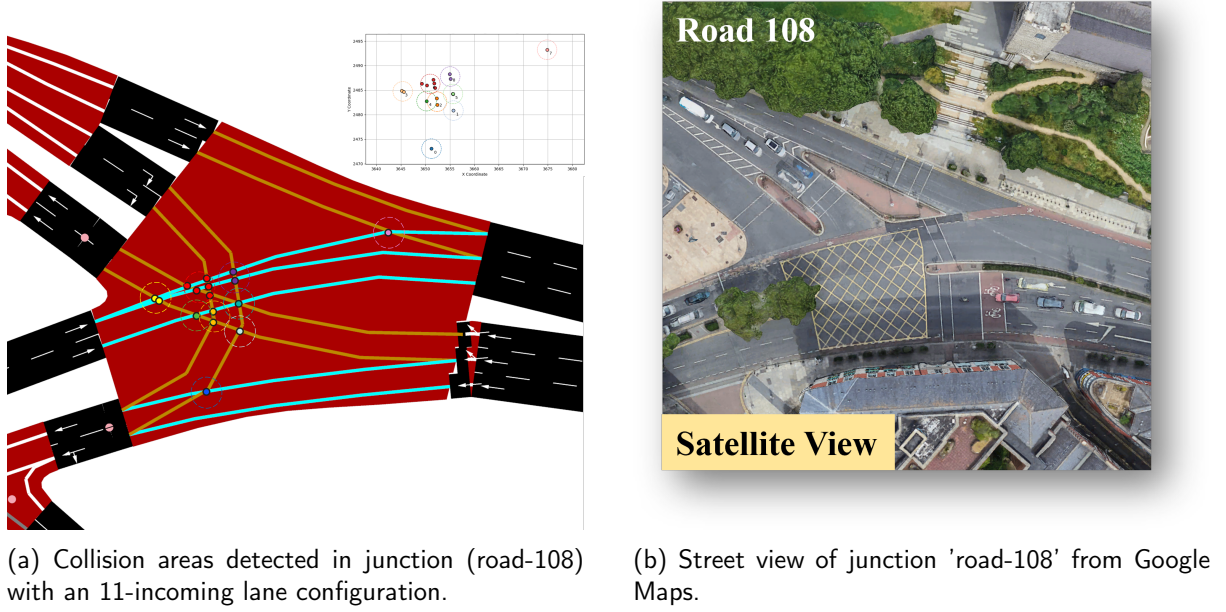


Figure 8.26: Comparison of detected collision areas and the real-world street view of junction 'road-108'.

A radius of ϵ meters (the *epsilon* parameter in DBSCAN) is selected as the threshold distance for clustering collision areas. This choice is based on the critical *Time-to-Collision* (*TTC*) threshold, which is typically considered 1.5 seconds for collision avoidance [Das and Maurya, 2019]. Given that a vehicle needs sufficient time to both enter and exit a collision area safely, the collision area radius is calculated using:

$$\epsilon = v_{\max} \times TTC - L_{\text{veh}}$$

where the maximum crossing speed in our implementation is $v_{\max} = 5$ m/s, and the average vehicle length is $L_{\text{veh}} = 5$ m and $TTC = 1.5$ s as a critical measurement. Substituting these values, the minimum required radius is 1.25 meters. To enhance safety while balancing collision overlap and crossing efficiency, we conservatively set the radius to 2 meters. This selection ensures that vehicles have sufficient clearance within the clustered collision areas, reducing the risk of conflicts while maintaining efficient intersection throughput.

The resulting collision areas of junction 'road-108', after applying DBSCAN, are depicted in Figure 8.26a. The post-clustering junction model represents an 11-incoming lane configuration and identifies 9 distinct collision areas, the corresponding street view is depicted in Figure 8.26b.

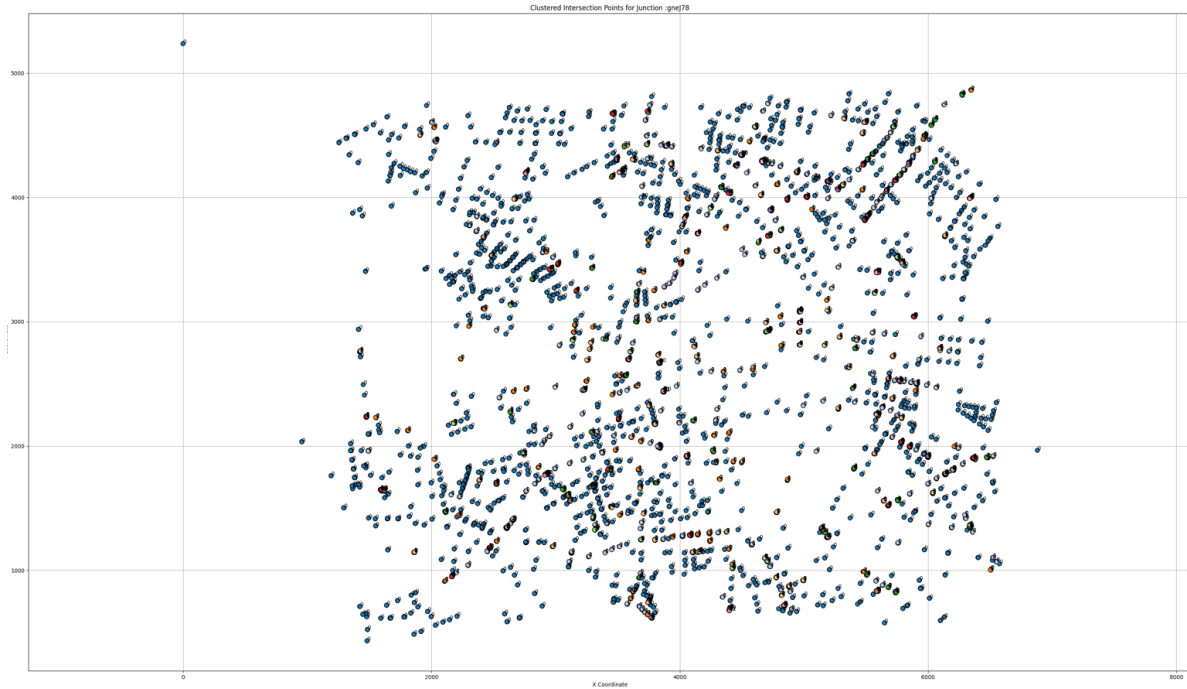


Figure 8.27: Automatically detected collision areas in Dublin city centre using DBSCAN clustering algorithm.

Collision areas in Dublin city centre are detected automatically by the proposed clustering method and are illustrated in Figure 8.27.

Evaluation of Zero-shot and Fine-tuning Approaches

Before evaluating the proposed Hierarchical Multi-Agent Reinforcement Learning (HRL-MAPPO) framework, it is essential to clarify the benchmark configurations for the baseline traffic-light control (TLC) scenarios. In the conventional TLC setting, only 30% of the vehicle fleet consists of connected and autonomous vehicles (CAVs), with the remainder being legacy human-driven vehicles. This configuration reflects a realistic near-term deployment stage, where partial automation coexists with conventional vehicles. In contrast, the TLC scenario assumes full CAV penetration (100%), denoted as TLC-F, thereby removing the variability and limitations inherent in human driving. This replacement inherently improves intersection coordination, reduces reaction time delays, and enables finer-grained signal control strategies. Nevertheless, even with 100% CAV penetration, congestion is not fully resolved, as vehicles remain subject to phase-based traffic-light control and the inherent optimisation constraints of the TLC paradigm.

In this study, all baseline and proposed optimization strategies were comprehensively evaluated over the full simulation period to assess their impact on network-wide performance. The TLC configuration, with partial CAV penetration, represents the least automated and most congestion-prone condition. The TLC-F, applying a 100% CAV penetration rate, improves coordination but remains constrained by signal plans. Learning-based baselines demonstrate further performance gains: the Parallel Node Monte Carlo Tree Search (PNMCTS) method applies a tree-search planning algorithm to schedule CAV flows dynamically, while the isolated PPO

(ISO-PPO) method employs reinforcement learning at individual intersections with further platoon length selection. The proposed Multi-Agent PPO (MAPPO) framework extends these capabilities by enabling decentralized agents to coordinate policies for global network optimization. In its zero-shot form, MAPPO achieves a substantial reduction in average travel delay (Figure 8.29) relative to both TLC (Figure 8.28) and other learning-based baselines, approaching near free-flow conditions in most urban areas. Nonetheless, persistent congestion remains along the Liffey River corridor, indicating the need for targeted fine-tuning in critical sub-network regions.

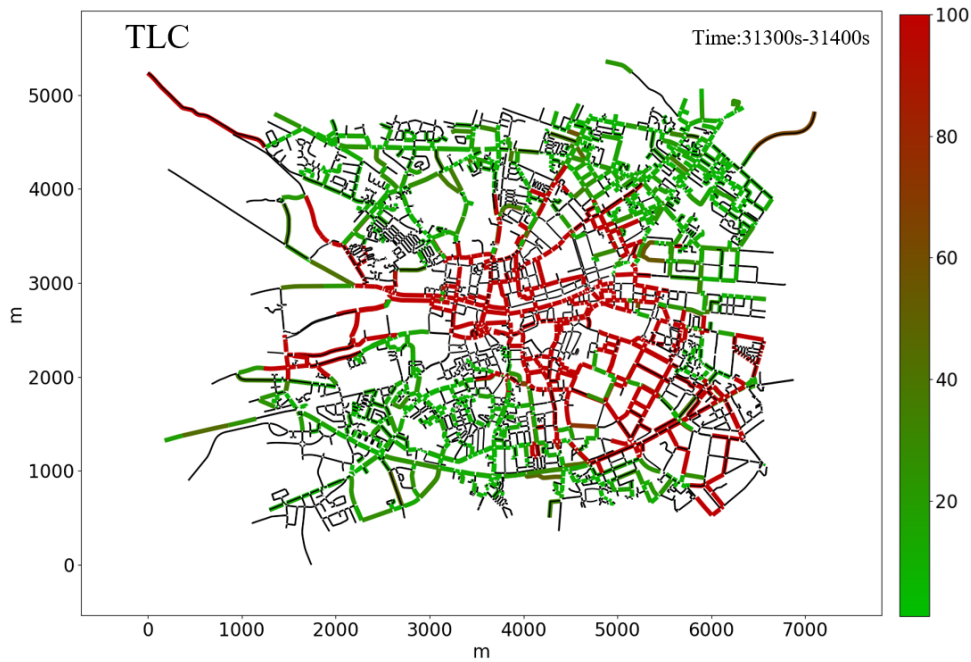


Figure 8.28: Traffic-light control on Dublin city network with 30% CAV penetration.

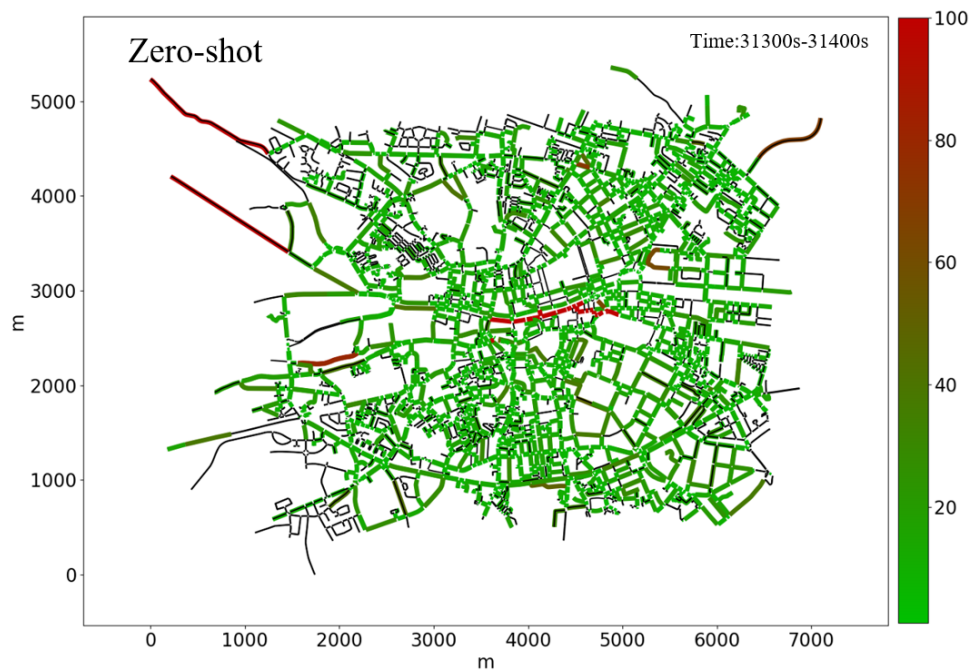


Figure 8.29: Proposed MAPPO zero-shot evaluation on Dublin city network.

To mitigate the heavy congestion, a sub-network centred on the Dublin City Council area was isolated for targeted fine-tuning (Figure 8.30). This sub-network contains 200 intersections in the most congested region of the city. The fine-tuning process further adapted the learned policy to local traffic dynamics, producing the results shown in Figure 8.31.

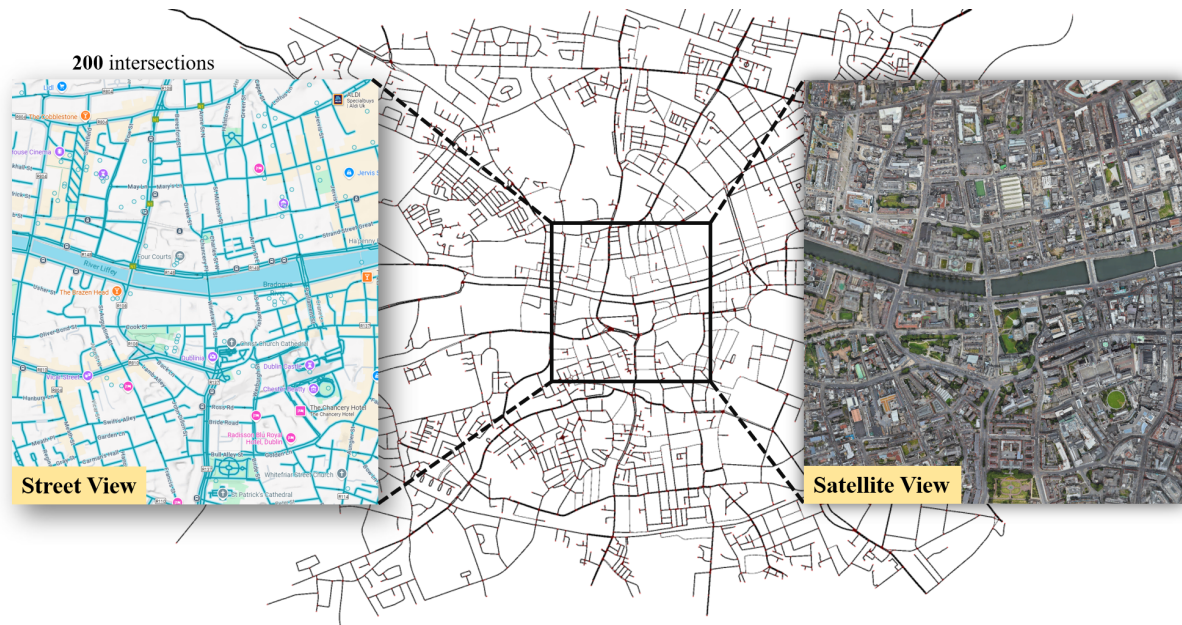


Figure 8.30: Fine-tune area in Dublin city network.

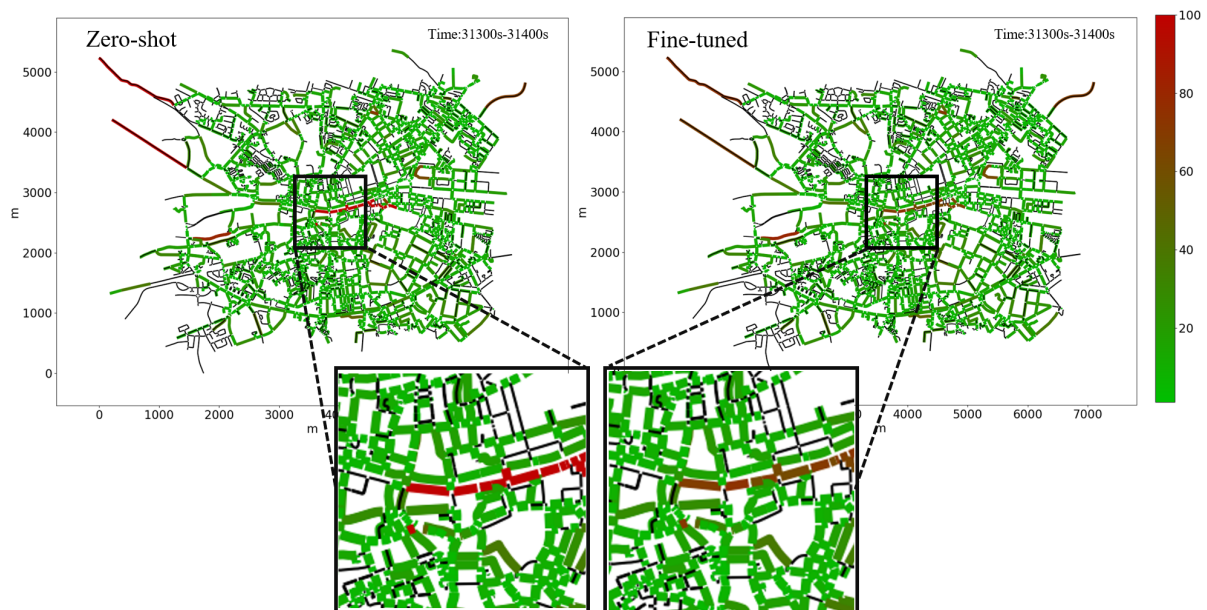


Figure 8.31: Proposed fine-tune evaluation on Dublin city network.

Table 8.6 summarises the comparative performance of all evaluated methods. The TLC baseline, with only 30% CAV penetration, records a TTR of 76.84%, indicating high variability and overall inefficiency. TT reaches 37,905 vehicles, with a TTD of 70,914.48 km, while ATT, WT, and WC stand at 238.47 s, 34.94 s, and 6.27, respectively. Replacing all legacy vehicles with CAVs in TLC-F yields notable improvements across all metrics: TTR decreases to 66.25%, TT rises to 42,254, and

Table 8.6: Comparison of Different Methods on Traffic Management Metrics. **Abbreviations:** TTR – Travel Reliability, TT – Total Throughput, TTD – Total Travel Distance, ATT – Average Travel Time, WT – Waiting Time, WC – Waiting Count.

Methods	TTR (%)	TT (veh)	TTD (km)	ATT (s)	WT (s)	WC
TLC	76.84	37 905	70 914.48	238.47	34.94	6.27
TLC-F	66.25	42 254	79 050.80	236.29	33.14	5.07
PNMCTS	20.13	47 332	88 550.96	230.07	26.78	4.93
ISO-PPO	18.76	47 541	88 941.97	229.47	25.64	3.85
MAPPO (Zero-shot)	12.03	47 727	93 291.90	225.27	20.13	2.01
MAPPO (Fine-tune)	9.49	48 025	93 964.82	223.78	18.27	2.00

TTD increases to 79,050.80 km, alongside marginal reductions in ATT, WT, and WC. These gains highlight the inherent efficiency benefits of full CAV penetration even under conventional traffic light control.

Among the learning-based baselines, PNMCTS reduces TTR to 20.13%, increases TT to 47,332, and lowers ATT to 230.07 s. ISO-PPO achieves a slightly better TTR of 18.76% and reduces ATT to 229.47 s, demonstrating the benefits of policy optimisation even without inter-agent coordination.

MAPPO in the zero-shot setting delivers a substantial leap in performance, with a TTR of 12.03%, the highest TT (47,727) and TTD (93,291.90 km) among the non-fine-tuned methods, and a reduced ATT of 225.27 s. WT and WC drop significantly to 20.13 s and 2.01, indicating smoother flow and fewer complete stops.

Fine-tuning MAPPO within the 200-intersection sub-network produces the best overall results: the lowest TTR (9.49%), highest TT (48,025) and TTD (93,964.82 km), and shortest ATT (223.78 s). WT and WC are further reduced to 18.27 s and 2.00, confirming that the hierarchical fine-tuning stage enables targeted local adaptation, further decreasing variability and congestion beyond the network-wide zero-shot optimisation.

Summary

The results highlight the need to differentiate between partial and full CAV penetration when defining baseline performance. Even without learning-based optimisation, transitioning from 30% to 100% CAV penetration (TLC to TLC-F) yields measurable gains in reliability, throughput, and delay reduction. However, these improvements are inherently limited by the constraints of traditional signal control. Among the learning-based methods, PNMCTS and ISO-PPO both deliver substantial reductions in TTR and improvements in ATT compared to TLC-F, demonstrating the effectiveness of policy optimisation through reinforcement learning and search-based planning.

MAPPO in its zero-shot configuration achieves a substantial improvement across all performance metrics, approaching near free-flow conditions in most parts of the network.

The fine-tuned MAPPO policy delivers the best overall results, with the lowest travel time variability, highest throughput, and minimal delays, confirming the value of hierarchical fine-tuning for addressing persistent localised bottlenecks. Overall, the evaluation validates the HRL-MAPPO

framework as a robust and scalable solution for adaptive traffic management in both mixed and fully autonomous fleets.

8.3.4 Ablation Study of HRL-MAPPO

Evaluation of Neighbouring Reward

First, we conduct an ablation study by varying the parameter α in the reward function:

$$R'_i = \alpha R_i + (1 - \alpha) \frac{1}{|C_i|} \sum_{j \in C_i} R_j \quad (8.1)$$

where α determines the trade-off between individual reward R_i and the mean reward of directly connected intersections. We examine multiple values of α , ranging from highly independent behaviour ($\alpha = 0.9$) to highly cooperative behaviour ($\alpha = 0.1$).

Figure 8.32 and 8.33 present the experimental results. Figure 8.32 illustrates the overall episode reward trends, while Figure 8.33 visualizes the per-agent episode rewards across different values of α .

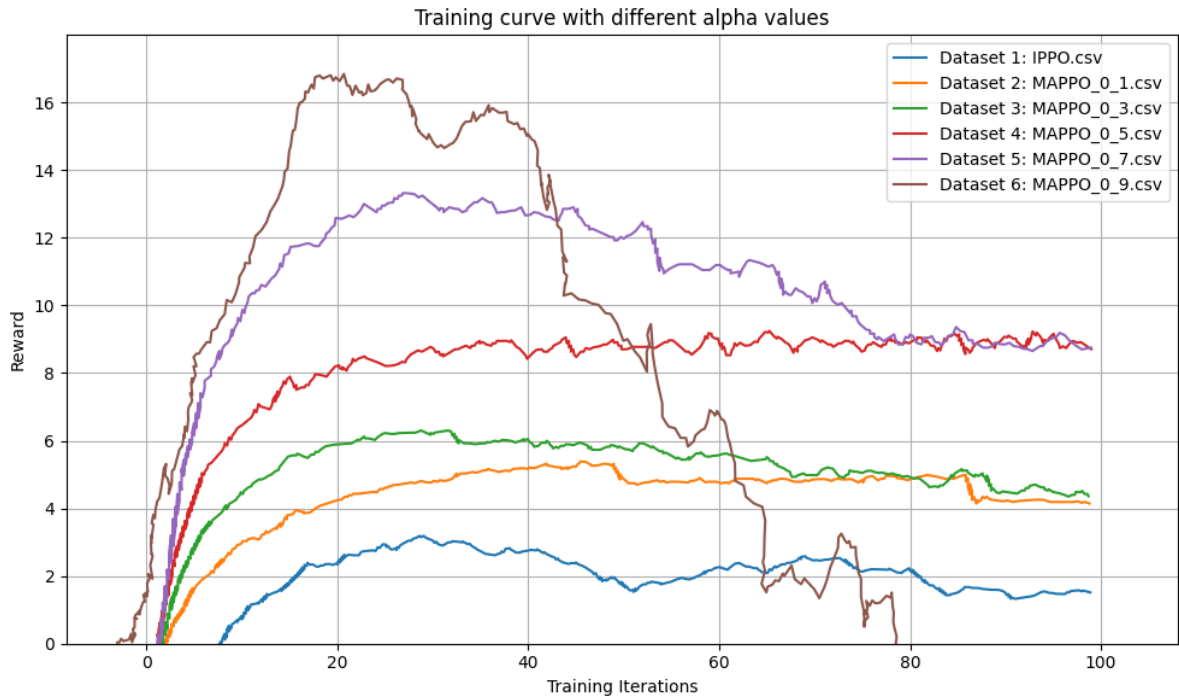
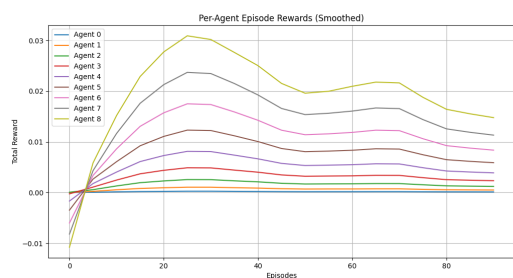


Figure 8.32: Overall episode rewards for different values of α . MAPPO refers to collaborative policies incorporating neighbour rewards at varying levels, controlled by parameter α , while ISO-PPO utilizes only individual observations and local rewards.

The results reveal that α plays a crucial role in shaping cooperative behaviour among intersections. ISO-PPO demonstrates that independent training leads to instability and uneven policy learning, the lack of collaboration results in suboptimal convergence and inefficient traffic management.

For $\alpha = 0.9$, agents focus primarily on local rewards with collaborative observation, achieving an



(a) ISO-PPO

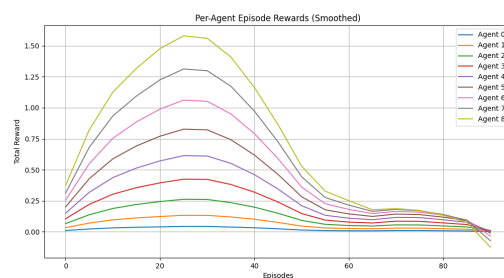
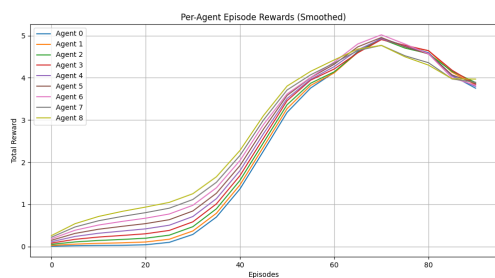
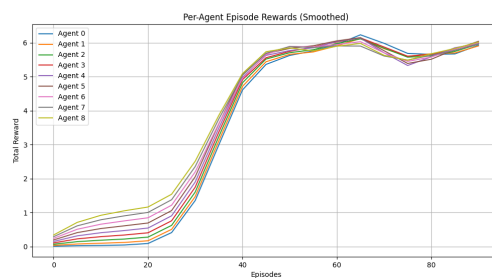
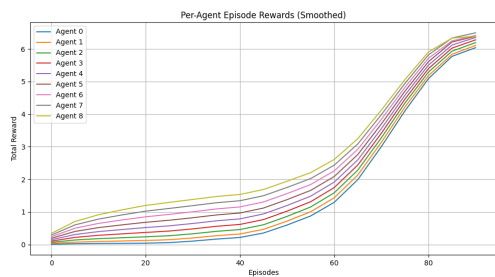
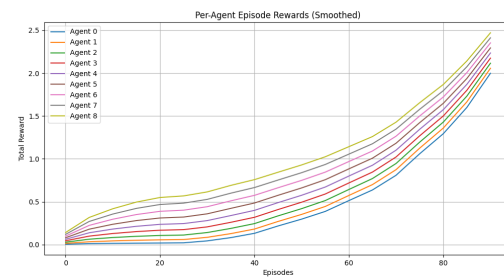
(b) $\alpha = 0.9$ (c) $\alpha = 0.7$ (d) $\alpha = 0.5$ (e) $\alpha = 0.3$ (f) $\alpha = 0.1$

Figure 8.33: Per-agent episode rewards for different values of α . MAPPO integrates cooperative observation with varying levels of neighbour rewards, while ISO-PPO relies solely on local observations and rewards.

initial performance boost. However, as training progresses, the rewards decline due to the emergence of non-cooperative policies that benefit early-stage learning but fail to achieve the long-term objective, leading to traffic congestion and inefficiency.

When $\alpha = 0.7$, agents begin to incorporate neighbouring rewards, fostering more stable learning dynamics. The overall performance improves as agents increasingly optimize their actions in a coordinated manner. However, performance degradation still persists, suggesting residual instability in policy convergence.

The optimal trade-off is observed at $\alpha = 0.5$, where agents achieve a balance between local optimization and cooperative decision making. This configuration results in smoother convergence, consistently higher episode rewards, and better system-wide traffic efficiency.

For $\alpha \leq 0.3$, excessive reliance on neighbour rewards leads to performance degradation. Training agents to rely on neighbour rewards in MARL disrupts individual learning incentives, leading to unstable training. The absence of self reward weakens the learning signal, making policy optimization difficult to reward feedback loops, where agents reinforce each other's actions without meaningful performance gains.

In summary, incorporating cooperative rewards enhances system stability and efficiency, but excessive dependency on neighbour rewards can be detrimental. The results underscore the importance of carefully tuning α to balance local decision-making with cooperative behaviour, ensuring optimal coordination in multi-agent traffic control systems.

Evaluation of Dynamic Platoon Selection

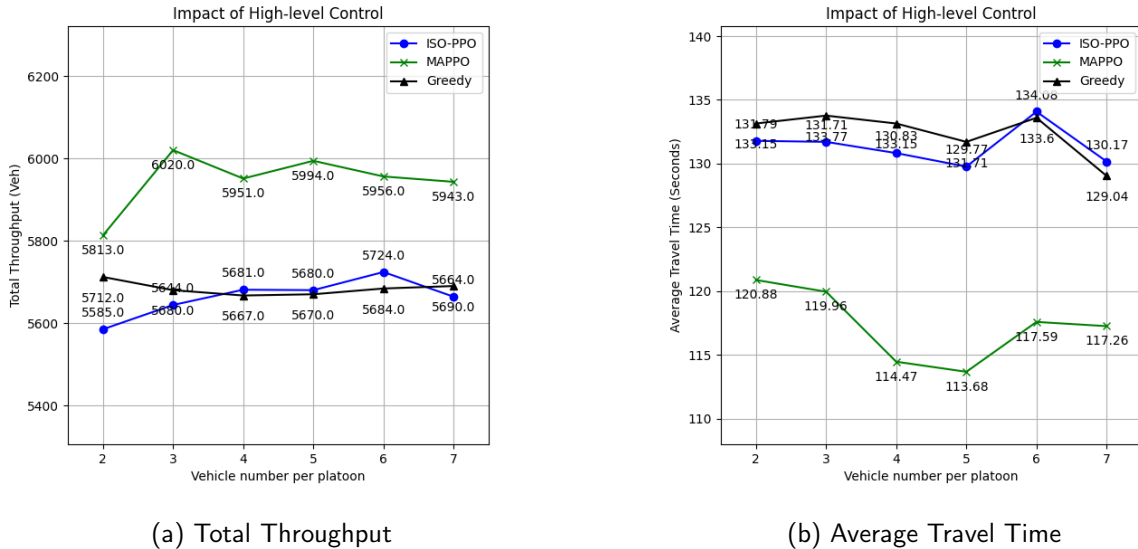


Figure 8.34: Comparison of ATT and TTP across different platoon lengths and control strategies.

Next, the study analyses the impact of different platoon lengths on the performance of various control strategies: Greedy, ISO-PPO, and MAPPO within a 3×3 homogeneous grid network with $\alpha = 0.5$. The Greedy policy permits platoons to cross with a maximum length defined by the value on the x-axis. For instance, if $x = 3$, only platoons with up to 3 vehicles are allowed to cross.

If a platoon exceeds this length, it is divided into smaller platoons to meet the length limit for the next pass. This process is repeated for different values of x , where the maximum allowable platoon length increases accordingly. In contrast, the proposed MAPPO approach incorporates the pressures of neighbouring intersections along with a collaborative reward, and training is performed collectively across all intersections using the centralized training and decentralised execution (CTDE) framework while inference is done locally.

Figure 8.34a shows the TTP achieved by the three control strategies as platoon length varies. The MAPPO method consistently achieves higher throughput compared to ISO-PPO and Greedy, indicating its ability to better utilize global pressure information from neighbouring intersections. For a platoon length of 2 vehicles, MAPPO maintains similar throughput of 5813 vehicles compared to 5712 for ISO-PPO and 5585 for Greedy. As the platoon length increases, the gap between MAPPO and the other methods increases, though MAPPO maintains a performance edge. This suggests that MAPPO's collaborative learning framework is particularly beneficial under constraints of longer platoons, where coordination is more dynamic.

Figure 8.34b presents the ATT (in seconds) across the three methods for varying platoon lengths. The results highlight that MAPPO achieves the lowest average travel time across all platoon lengths, demonstrating its efficiency in reducing delay. For a platoon length of 2, MAPPO records an average travel time of 120.88 seconds, significantly lower than Greedy's 133.15 seconds and ISO-PPO's 131.79 seconds. The performance advantage of MAPPO is maintained as platoon lengths increase, with MAPPO achieving an average travel time of 117.26 seconds at a platoon length of 7, compared to 129.04 seconds for ISO-PPO and 130.17 seconds for Greedy.

We can conclude that without information sharing from neighbouring intersections, PPO achieves a crossing strategy similar to the Greedy approach. This is because the objective of individual agents is to minimize queuing at their own intersections, leading to localized optimization. However, with MAPPO, the inclusion of information from neighbouring intersections enables a more coordinated strategy. By considering both the local and neighbouring pressures, MAPPO collectively learns to balance traffic flow across the network, reducing congestion and improving overall throughput and travel time.

8.4 Analysis and Insights

While the proposed framework demonstrates strong performance in optimizing traffic scheduling, the complexity of the problem domain introduces several fundamental challenges. Unlike traditional applications of MCTS in structured and finite environments such as Chess or Go, traffic scheduling operates within an unbounded and dynamically evolving state-action space. These characteristics pose inherent difficulties for policy training, model generalization, and real-time decision making. Analysing these challenges provides valuable insights into the limitations of current approaches and informs potential directions for future enhancement.

State-Action Space Complexity: In classic MCTS applications such as Go or Chess, the search (action) space is constrained, with a limited number of legal moves at each step—362 possible moves in Go and up to 218 moves in Chess. Moreover, this space gradually shrinks as the game

progresses, reducing the number of available actions over time.

In traffic scheduling, the number of possible actions at each intersection is significantly larger than in conventional planning domains due to the continuous nature of vehicle movements. Furthermore, in scenarios where platoons are still crossing the intersection, the action space typically does not reduce, maintaining a high level of complexity throughout decision making. This renders exhaustive search computationally infeasible.

To address this challenge, we discretize the action space into 640 predefined actions. This approach ensures computational tractability and enables real-time scheduling. However, it introduces limitations in capturing fine-grained control over CAV behaviours, which may result in sub-optimal crossing schedules in some cases. Nonetheless, the reduced action space offers a practical trade-off between scheduling quality and computational efficiency.

Alternative Approaches for Large Action Spaces: Recent advancements in policy optimization have led to the development of Group Relative Policy Optimization (GRPO), which has demonstrated strong performance in large-scale decision-making tasks, including mathematical problem-solving and large language model pretraining [Shao et al., 2024]. Unlike conventional policy optimization methods, GRPO removes the dependency on a value network and instead leverages Z-score normalization within a sampling group to estimate advantages. This approach significantly reduces training overhead while maintaining competitive performance. Incorporating such techniques into traffic scheduling could offer a promising direction for managing large and high-dimensional action spaces more efficiently.

Scalability Limitations: The current neural network design supports a board containing maximum of 8 platoons and 8 collision areas. Expanding networks size do not always yield better traffic efficiency due to the increasing difficulty of synchronizing traffic flow across multiple intersections. As the network size increases, uncertainty, coordination complexity, and error propagation accumulate, could result in suboptimal scheduling. Identifying the optimal granularity for intersection control remains an open challenge [Gu et al., 2024].

Value Network Generalization Issues: Unlike Chess, where the number of possible initial states is vast but still finite, traffic scheduling presents a continuous and highly stochastic initial state space due to variable vehicle arrivals. In Chess, deep learning models, such as those used in AlphaZero, can learn strong heuristics through self-play and MCTS, but they still rely on search rather than perfect generalization across all states. In contrast, traffic scheduling involves a virtually infinite number of possible states and high-dimensional, non-deterministic interactions, making it significantly harder for a value network to accurately estimate unseen situations.

Traditional MCTS mitigates this issue by performing extensive rollouts, allowing the search to refine decisions beyond the network's raw estimates. However, in large-scale traffic scheduling, exhaustive rollouts are computationally prohibitive due to the complexity of the decision space. To address this, a short tree search is integrated to the trained policy, refining decisions dynamically instead of relying solely on the value network's approximation. This hybrid approach ensures that scheduling decisions are informed by both learned patterns and real-time search, mitigating over-reliance on function approximation, which remains a key bottleneck in high-dimensional traffic environments.

MCTS Exploration-Exploitation Balance: The exploration constant C in MCTS plays a crucial role in determining the balance between exploration and exploitation. A higher C encourages broader exploration of the search tree, while a lower C promotes early exploitation of high-reward paths. This is conceptually similar to the temperature parameter in softmax-based policies. In our training setup, a higher temperature is used during early training phases to encourage diverse exploration, while top-K sampling prevents impractical action selection by restricting the search space to a subset of promising decisions. This dynamic tuning of exploration-exploitation helps stabilize learning in high-dimensional spaces.

Pruning Tree Growth: Unlike in Chess or Go, where the number of legal moves per turn decreases over time, the number of possible actions in an intersection remains constant due to dynamic traffic arrival patterns. Fixed-depth pruning may eliminate promising child nodes prematurely, while expanding all possible actions is computationally infeasible. The current implementation selectively expands nodes with high estimated return rather than uniformly growing the tree. This strategy prioritizes promising actions while avoiding unnecessary branching. An alternative approach is **progressive widening**, which incrementally expands the action space as more simulations are performed. Instead of fully enumerating all actions upfront, it controls tree branching by gradually adding new child nodes based on visitation count thresholds. This technique has proven effective in large or continuous action spaces and can balance the exploration-exploitation trade-off efficiently. Incorporating progressive widening into the intersection scheduling tree search could further reduce computational burden while maintaining coverage of high-value trajectories, particularly under high traffic variance or sparse reward signals [Sunberg and Kochenderfer, 2018, Lim et al., 2021].

Stabilizing Learning with Regularization is Important: In deep RL, instability often arises from overfitting to high-reward trajectories while neglecting diverse exploration. To enhance stability, dropout regularization is applied to prevent over reliance on specific pathways. Noise injection in value predictions encourages policy robustness, ensuring that small fluctuations and noise in training data do not lead to drastic changes in decisions. Additionally, batch normalization is incorporated to maintain smooth gradient propagation and avoid learning instability.

Challenges in Multi-Agent Coordination: MAPPO introduces additional complexities beyond single-agent reinforcement learning. One major challenge is coordination complexity, as MAPPO is inherently designed for agents that interact closely. The assumption that multi-agent learning always outperforms independent agents does not hold in all cases. In some scenarios, increased coordination introduces unnecessary constraints, leading to suboptimal solutions. This is primarily caused by non-stationarity, where agents continuously update their policies, making the environment highly dynamic and unstable. Exploration conflicts arise when different agents prioritize conflicting objectives, leading to inefficient scheduling. Additionally, sparse rewards make learning difficult, as delayed feedback does not provide strong learning signals.

Instead of relying on global information exchange, this study introduces a neighbour-based traffic optimization reward model using sub-lane traffic pressure. Intersection controllers share detailed pressure information from approaching lanes while incorporating both self-rewarding and neighbour-rewarding into the final reward function. This ensures that each agent receives a sufficient reward signal for effective collective decision-making without requiring extensive

inter-agent communication.

To further enhance coordination while maintaining scalability, the proposed approach leverages **CTDE**. During training, a centralized critic accesses global traffic state information to optimize policy updates, allowing agents to learn an implicit coordination strategy. However, during execution, each agent relies only on its local observations and reward signals, ensuring real-time decision-making without communication overhead.

By integrating CTDE, agents develop policies that generalize well across varying traffic conditions, striking a balance between decentralized efficiency and cooperative learning. This approach enables robust adaptation in large-scale traffic networks, where direct inter-agent coordination becomes infeasible due to latency constraints and computational complexity.

8.5 Summary

This chapter presents a comprehensive evaluation of the proposed hierarchical multi-agent traffic management framework, with a focus on addressing the central research question:

Can a hierarchical, multi-agent reinforcement learning approach effectively optimize traffic flow in large-scale heterogeneous unsignalized urban traffic networks? And how does it compare to existing methods in terms of throughput, travel time, and travel-time reliability?

The evaluation is structured through a series of detailed analyses, each targeting a critical dimension of the research question.

- **Hierarchical Structure for Decentralized Traffic Optimization:** The proposed framework mitigates coordination complexity by hierarchically decomposing control tasks. At the mid-level, PNMCTS-based platoon scheduling enhances local decision robustness, significantly reducing intersection-level delay. At the top level, MAPPO-based coordination among decentralized agents improves throughput (TT) while reducing average travel time (ATT) and total time ratio (TTR), contributing to scalable and efficient global control.
- **Heterogeneous Intersection Representation and Collision Area Detection:** The discrete-time state-space model (DTSSM) standardizes heterogeneous intersection geometries and dynamics, reducing uncertainty arising from layout diversity. Additionally, dynamic collision area identification via DBSCAN clustering enables adaptive synchronization, localizing coordination efforts to high-interaction zones and reducing complexity induced by geometric and structural variability.
- **Balancing Short-Term and Long-Term Traffic Objectives:** The hierarchical reward design effectively reconciles local intersection efficiency with long-term network-wide performance. By propagating informative and temporally consistent reward signals upward, the framework facilitates high-level decision making on platoon selection, demonstrating that structured reward decomposition is critical for hierarchical traffic optimization.
- **Scalability and Training Strategies for Large-Scale Multi-Agent Systems:** Extensive experiments on large-scale networks, including a 1,645-intersection layout based on Dublin,

validate the scalability of the hierarchical MAPPO approach. Collision-area-based agent clustering reduces training and inference complexity, allowing agents to converge towards globally efficient policies via shared neighbouring rewards. Zero-shot deployment followed by fine-tuning further demonstrates the adaptability and generalization capacity of the proposed HRL-MAPPO framework.

In summary, the evaluation substantiates that hierarchical control decomposition, standardized intersection representation, and scalable multi-agent coordination substantially mitigate the negative impacts of network scaling. These results offer actionable insights for the development of robust, future-ready urban traffic management systems.

9 Conclusion

Urban traffic congestion remains a critical challenge for modern cities, requiring innovative and scalable solutions to improve reliability, efficiency, and safety. This thesis has explored the potential of hierarchical and decentralized multi-agent deep reinforcement learning (MADRL) for optimizing large-scale unsignalized urban traffic networks of connected and automated vehicles (CAVs). By integrating hierarchical reinforcement learning (HRL), multi-agent reinforcement learning (MARL), and fine-grained scheduling algorithms, this research has developed a framework capable of managing automated traffic flow efficiently across diverse intersection geometries and dynamic traffic conditions at scale.

In answering the research questions posed in Chapter 1, this chapter summarizes the key findings and contributions of this research, discusses its limitations, and outlines potential directions for future work. The framework introduces several innovative techniques, including decentralized throughput optimization with Multi-Agent Proximal Policy Optimization (MAPPO), a standardized intersection representation with automated collision detection, an optimized platoon scheduling algorithm using Parallel Neural Monte Carlo Tree Search (PNMCTS), and a zero-shot and few-shot fine-tuning approach to enhance policy generalization. While the framework demonstrates promising results in simulation, several challenges remain in ensuring its robustness, scalability, and real-world applicability.

9.1 Summary of Technical Approach

This section summarizes the key technical approaches adopted to addressing the challenge of optimizing large-scale, heterogeneous unsignalized urban traffic networks.

9.1.1 Hierarchical Reinforcement Learning Framework for Urban Traffic Management

The foundation of our solution is a novel hierarchical framework that systematically decomposes the problem of large-scale urban traffic optimization into multiple layers, addressing scalability, adaptability, and computational efficiency challenges. The hierarchical structure introduces three levels of decision making:

- **High-Level Throughput Optimization:** A decentralized MARL approach based on MAPPO is employed to manage traffic throughput (based on platoon size selection) at a network-wide scale, utilizing neighbouring rewards and sub-lane traffic pressure as key mechanisms.

- **Mid-Level Scheduling:** PNMCTS was developed to generate optimized crossing schedules for vehicle platoons, delivering near-optimal and conflict-free intersection management.
- **Low-Level Vehicle Trajectory Control:** A multi-segment acceleration control strategy was implemented to synchronize CAV movements based on the optimized crossing schedules.

The hierarchical integration of these layers enables a scalable and decentralized framework for urban traffic optimization. By incorporating collision avoidance, automated intersection transformation, and real-time adaptive decision-making, the framework provides feasible and safe vehicle crossings under diverse traffic conditions.

9.1.2 Decentralized Multi-Agent Reinforcement Learning with MAPPO for Traffic Coordination

A major contribution of this research is the development of a MARL-based approach for urban traffic optimization leveraging MAPPO, which enables decentralized training with cooperative execution. Each intersection functions as an autonomous RL agent, making localized traffic control decisions based on real-time observations while indirectly coordinating with neighbouring intersections via a novel sub-lane **traffic pressure** signal. This decentralized MAPPO framework offers:

- **Scalability:** By allowing intersections to operate independently, the approach mitigates the computational bottlenecks of centralized RL solutions.
- **Robustness to Dynamic Traffic Conditions:** The framework dynamically adjusts intersection policies based on real-time platoon arrivals.
- **Reduced Coordination Overhead:** Unlike fully cooperative MARL approaches that require explicit inter-agent communication, neighbouring intersection optimization provides an indirect yet effective coordination mechanism.

9.1.3 Standardized Intersection Representation with Automated Collision Detection

One of the major obstacles in optimizing large-scale unsignalized traffic networks is the heterogeneity of intersection geometries. This research introduces a novel *intersection transformation model* based on a **discrete-time state space representation (DTSSM)** to standardize intersection geometries and automate collision detection. Key feature of this model include:

- Transforming the intersection scheduling problem into a structured **board-game-like decision process** via DTSSM, enabling learning and optimization to proceed.
- **Automated conflict zone detection mechanism** that recognizes and merges conflict zones, functioning effectively across **diverse and complex intersection topologies** without requiring manual feature engineering.

9.1.4 Platoon Scheduling with PNMCTS

This research introduces a novel PNMCTS algorithm for platoon scheduling at unsignalized intersections. The key challenge in unsignalized intersection management is to schedule platoon movements efficiently while ensuring safety and minimizing delays. This thesis addresses this challenge by:

- Enhancing standard Neural Monte Carlo Tree Search to suit the specific board scheduling game.
- Utilizing the **Upper Confidence Bound (UCB)** function to balance exploration and exploitation, thereby improving decision-making efficiency under uncertainty.
- Employing **parallelization techniques** to enhance computational efficiency, making the method suitable for large-scale training.

9.1.5 Zero-Shot and Few-Shot Fine-Tuning

This research incorporates a **geometry-agnostic training method** to handle the heterogeneity present in real-world urban networks. The method transforms intersections with varying geometries into a unified representation, effectively decoupling intersection geometry from the training process. The transformed representation is then fed into a global network for training. The key challenge is that optimizing thousands of heterologous intersections using one single network may not be efficient enough. This thesis addresses the challenge by:

- **Zero-Shot Evaluation:** Enabling policy evaluation in new intersection geometries without any additional training, which can enhance traffic efficiency across most areas of the urban network by relying purely on learned experience.
- **Few-Shot Fine-Tuning:** Implementing fine-tuning in highly congested areas using fresh traffic data to enhance policy adaptability, allowing the model to better adjust to changing traffic patterns and improve overall traffic efficiency.

9.2 Answering the Research Questions

This section considers how each of the research questions posed in Chapter 1 has been answered by this work beginning with the overarching research question - **can a hierarchical, multi-agent reinforcement learning approach effectively optimize traffic flow in large-scale heterogeneous unsignalized urban traffic networks? Specifically, does this approach improve traffic flow in terms of throughput, travel time, and travel-time reliability compared to existing methods?**

This thesis introduced a hierarchical MARL-based framework that significantly improves traffic flow in large-scale, heterogeneous unsignalized urban networks. Compared to traditional actuated traffic light control, our approach enhances system performance in terms of throughput, average travel time, and travel-time reliability. Specifically, this thesis introduces two modes of deployment: a zero-shot generalization method that improves traffic conditions across most regions without

additional training, and a few-shot fine-tuning method that further optimizes performance in heavily congested areas. Experimental results in simulation demonstrate improvements of 125%, 5%, and 85% in throughput, average travel time, and travel-time reliability, respectively, over traffic light control. With fine-tuning in congested zones, additional gains of 0.6%, 0.8%, and 23% are observed, respectively (Section 8.3.3).

The answers to the sub-research questions are presented as follows:

1. **What is the appropriate hierarchical structure for a decentralized traffic management framework?** This thesis introduces a three-level hierarchical structure to decompose traffic control across intersections. At the top level, a system-wide optimization agent monitors network-level traffic pressure and dynamically determines optimal platoon lengths. These decisions are passed to the mid-level agents responsible for intersection scheduling, which focus solely on resolving local crossing conflicts and providing feedback to the top level via reward signals. The successful schedules are then handed to the low-level agents, which compute precise trajectories for CAVs, ensuring accurate execution of the crossing plan. This hierarchical design has been extensively evaluated on both single-intersection scenarios (Section 8.3.1) and 3×3 grid networks, achieving state-of-the-art performance (Section 8.3.2).
2. **How can representation of intersection geometries be standardized?** This thesis provides a standardized representation of intersection geometries in three stages. First, intersection dynamics are modelled using the DTSSM framework. Second, these dynamics are transformed into a fixed-size board game representation, where busy intersections produce denser boards and sparse boards represent less active ones. This uniform structure facilitates consistent training and inference using PNMCTS, enabling generalization to unseen geometries as long as they fit within the chosen board dimensions. Third, the transformation is adapted to real-world intersections using DBSCAN to automatically identify conflict zones, thereby enabling automated board construction. This representation underpins the entire framework and has been validated through extensive city-scale experiments described in Chapter 8.
3. **How can the framework balance short-term individual goals with long-term traffic network objectives?** To address the balance between local and global objectives, this thesis introduces a hierarchical reward decomposition strategy. Specifically, the top-level and mid-level agents are coupled through a level-specific reward propagation mechanism. Intersection scheduling decisions at the mid-level influence system-wide optimization by providing feedback to the high-level agent, thus aligning local scheduling quality with global traffic performance. The robustness of the proposed PNMCTS-based intersection scheduling is thoroughly evaluated through ablation studies, demonstrating superior performance over FIFO policies on single unsignalized intersections (Section 8.2.1), as well as outperforming both traditional and RL-based traffic signal controllers in grid network scenarios (Section 8.2.2).
4. **What are effective strategies for training hierarchical multi-agent systems for large-scale, heterogeneous urban networks?** This thesis adopts MAPPO for top-level,

network-wide coordination, leveraging neighbouring observations and rewards to encourage cooperation among agents in adjacent areas. For mid-level scheduling, PNMCTS is proposed to resolve local crossing conflicts effectively. The HRL-MAPPO framework is thoroughly evaluated through ablation studies on cooperative reward design and dynamic platoon length selection in Section 8.3.4. Subsequently, the framework is tested on a realistic Dublin city network using both zero-shot generalization and few-shot fine-tuning strategies. Experimental results show that HRL-MAPPO achieves fast convergence, maintains training stability, and significantly outperforms traditional traffic light control methods in both synthetic and real-world scenarios.

In conclusion, this thesis answers the overarching research question positively by demonstrating that a hierarchical MARL-based approach can effectively optimize traffic flow in large-scale, heterogeneous, unsignalized urban networks. Each sub-question is systematically tackled: the framework’s hierarchical structure ensures scalable coordination, the standardized intersection representation enables generalization, and the reward decomposition mechanism balances local and global objectives. Extensive experiments confirm the framework’s effectiveness, robustness, and potential practical applicability, validating it as a promising solution for intelligent traffic management.

9.3 Limitations and Future Work

While this study presents a promising hierarchical MARL framework for intelligent traffic management, several limitations and future research directions merit consideration:

- **Assumption of Full CAV Adoption:** The current framework assumes complete deployment of CAVs, which may not reflect near-term traffic conditions. In realistic scenarios with mixed traffic, the coexistence of human-driven and autonomous vehicles poses significant challenges. Future work should explore hybrid RL strategies capable of dynamically adapting to varying CAV penetration rates.
- **Reliance on Perfect Communication and Sensing:** The effectiveness of the framework depends on high-fidelity sensing and low-latency communication. In practice, network delays, sensor noise, and data loss can degrade performance. Future research should incorporate robustness mechanisms such as uncertainty-aware decision-making, fault-tolerant control, and federated learning to enhance reliability under imperfect conditions.
- **Policy Generalization Across Irregular Intersection Geometries:** The intersection transformation model facilitates generalization across typical layouts, but its adaptability to irregular or complex geometries remains limited. The current fixed 2-meter merging condition could be replaced by learnable, geometry-aware collision avoidance strategies.
- **Scheduling Constraints Due to Fixed Board Size in PNMCTS:** To balance scheduling performance with computational efficiency, the PNMCTS module currently restricts the board size to $8 \times 8 \times 2$, limiting its applicability to larger intersections. Future work should

investigate adaptive board resizing, RL-assisted node expansion, and more efficient tree search strategies to enhance scalability without incurring prohibitive computational cost.

- **Limited Consideration of Dynamic Demand and Multi-Modal Traffic:** The current framework primarily targets vehicular throughput optimization under relatively stable demand. Future research should extend the model to account for dynamic traffic patterns for longer traffic simulation and support multi-modal transportation systems, including pedestrians, cyclists, and public transit, to provide a more comprehensive urban mobility solution.

Bibliography

- [Abbas-Turki et al., 2023] Abbas-Turki, A., Mualla, Y., Gaud, N., Calvaresi, D., Du, W., Lombard, A., Dridi, M., and Koukam, A. (2023). Autonomous intersection management: Optimal trajectories and efficient scheduling. *Sensors (Basel, Switzerland)*, 23.
- [Abdoos and Bazzan, 2021] Abdoos, M. and Bazzan, A. L. (2021). Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. *Expert systems with applications*, 171:114580.
- [Ahn and Del Vecchio, 2016] Ahn, H. and Del Vecchio, D. (2016). Semi-autonomous intersection collision avoidance through job-shop scheduling. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 185–194.
- [Al-Sharman et al., 2023] Al-Sharman, M., Dempster, R., Daoud, M. A., Nasr, M., Rayside, D., and Melek, W. (2023). Self-learned autonomous driving at unsignalized intersections: A hierarchical reinforced learning approach for feasible decision-making. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):12345–12356.
- [Al-Sharman et al., 2024] Al-Sharman, M., Edes, L., Sun, B., Jayakumar, V., Daoud, M. A., Rayside, D., and Melek, W. (2024). Autonomous driving at unsignalized intersections: A review of decision-making challenges and reinforcement learning-based solutions. *arXiv preprint arXiv:2409.13144*.
- [Albeaik et al., 2022] Albeaik, S., Bayen, A., Chiri, M. T., Gong, X., Hayat, A., Kardous, N., Keimer, A., McQuade, S. T., Piccoli, B., and You, Y. (2022). Limitations and improvements of the intelligent driver model (idm). *SIAM Journal on Applied Dynamical Systems*, 21(3):1862–1892.
- [Ali et al., 2020] Ali, H. A., Khaled, S., Shehata, O. M., and Morgan, E. I. (2020). Decentralized intersection management of autonomous vehicles using nonlinear mpc. *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 410–415.
- [Almobayedh, 2019] Almobayedh, H. B. (2019). Simulation of the impact of connected and automated vehicles at a signalized intersection. Master’s thesis, University of Dayton.
- [Amodei et al., 2016] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

- [Andert et al., 2017a] Andert, E., Khayatian, M., and Shrivastava, A. (2017a). Crossroads: Time-sensitive autonomous intersection management technique. In *Proceedings of the 54th Annual Design Automation Conference 2017, DAC '17*, page 1–6. ACM.
- [Andert et al., 2017b] Andert, E. P., Khayatian, M., and Shrivastava, A. (2017b). Crossroads: Time-sensitive autonomous intersection management technique. *Proceedings of the 54th Annual Design Automation Conference 2017*.
- [Antonio, 2022] Antonio, Guillen-Perez; Maria-Dolores, C. (2022). Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections. *IEEE Transactions on Vehicular Technology*, 71(7):7033–7043.
- [Antonio and Maria-Dolores, 2022] Antonio, G.-P. and Maria-Dolores, C. (2022). Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections. *IEEE Transactions on Vehicular Technology*, 71(7):7033–7043.
- [Aoki and Rajkumar, 2018] Aoki, S. and Rajkumar, R. (2018). Dynamic intersections and self-driving vehicles. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, page 320–330. IEEE.
- [Au and Stone, 2010] Au, T.-C. and Stone, P. (2010). Motion planning algorithms for autonomous intersection management. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*.
- [Ayeelyan et al., 2022] Ayeelyan, J., Lee, G.-D., Hsu, H.-C., and Hsiung, P.-A. (2022). Advantage actor-critic for autonomous intersection management. *Vehicles*.
- [Azimi et al., 2014] Azimi, R., Bhatia, G., Rajkumar, R. R., and Mudalige, P. (2014). Stip: Spatio-temporal intersection protocols for autonomous vehicles. In *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE.
- [Azimi et al., 2013] Azimi, S. R., Bhatia, G., Rajkumar, R., and Mudalige, P. (2013). Reliable intersection protocols using vehicular networks. In *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems, ICCPS '13*, page 1–10. ACM.
- [Ba and Tordeux, 2023] Ba, I. and Tordeux, A. (2023). Signalized and unsignalized road traffic intersection models: A comprehensive benchmark analysis. *Collective Dynamics*, 8:1–21.
- [Bacon et al., 2017] Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 1726–1734. AAAI Press.
- [Barron and Ishii, 1989] Barron, E. and Ishii, H. (1989). The bellman equation for minimizing the maximum cost. *NONLINEAR ANAL. THEORY METHODS APPLIC.*, 13(9):1067–1090.
- [Bashiri and Fleming, 2017] Bashiri, M. and Fleming, C. H. (2017). A platoon-based intersection management system for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 667–672. IEEE.

- [Bashiri et al., 2018] Bashiri, M., Jafarzadeh, H., and Fleming, C. H. (2018). Paim: Platoon-based autonomous intersection management. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, page 374–380. IEEE.
- [Bazzan, 2009] Bazzan, A. L. C. (2009). Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(1):52–100.
- [Behrisch et al., 2011] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo—simulation of urban mobility: An overview. In *Proceedings of the Third International Conference on Advances in System Simulation*, pages 63–68.
- [Belkhouche, 2019] Belkhouche, F. (2019). Collaboration and optimal conflict resolution at an unsignalized intersection. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2301–2312.
- [Bian et al., 2019] Bian, Y., Li, S. E., Ren, W., Wang, J., Li, K., and Liu, H. X. (2019). Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control. *IEEE Transactions on Industrial Electronics*, 67(12):10744–10754.
- [Bian et al., 2020] Bian, Y., Li, S. E., Ren, W., Wang, J., Li, K., and Liu, H. X. (2020). Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control. *IEEE Transactions on Industrial Electronics*, 67:10744–10754.
- [Bisht and Shet, 2020] Bisht, N. and Shet, R. A. (2020). Platoon-based cooperative intersection management strategies. *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–6.
- [Board, 2010] Board, T. R. (2010). *Highway Capacity Manual 2010*. National Academies Press.
- [Borges et al., 2021] Borges, D. F., Leite, J. P. R., Moreira, E. M., and Carpinteiro, O. A. (2021). Traffic light control using hierarchical reinforcement learning and options framework. *IEEE Access*, 9:99155–99165.
- [Botvinick, 2012] Botvinick, M. M. (2012). Hierarchical reinforcement learning and decision making. *Current opinion in neurobiology*, 22(6):956–962.
- [Browne et al., 2012] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.
- [Buzachis et al., 2020] Buzachis, A., Celesti, A., Galletta, A., Fazio, M., Fortino, G., and Villari, M. (2020). A multi-agent autonomous intersection management (ma-aim) system for smart cities leveraging edge-of-things and blockchain. *Information Sciences*, 522:148–163.
- [Canese et al., 2021] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., and Spanò, S. (2021). Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11):4948.

- [Carlino et al., 2013] Carlino, D., Boyles, S. D., and Stone, P. (2013). Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 529–534. IEEE.
- [Cederle et al., 2024] Cederle, M., Fabris, M., and Susto, G. A. (2024). A distributed approach to autonomous intersection management via multi-agent reinforcement learning. *arXiv preprint arXiv:2405.08655*.
- [Chen et al., 2020] Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., and Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3414–3421.
- [Chen et al., 2018] Chen, J., Wang, Z., and Tomizuka, M. (2018). Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1239–1244.
- [Chu et al., 2019] Chu, T., Wang, J., Codecà, L., and Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095.
- [Cong et al., 2022] Cong, X., Yang, B., Gao, F., Chen, C., and Tang, Y. (2022). Virtual Platoon based CAVs Cooperative Driving at Unsignalized Intersection. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 53–58. IEEE.
- [Cowling et al., 2012] Cowling, P. A., Ward, C. D., and Powley, E. J. (2012). Monte carlo tree search: Long term versus short term planning. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE.
- [Dai et al., 2016] Dai, P., Liu, K., Zhuge, Q., Sha, E. H.-M., Lee, V. C. S., and Son, S. H. (2016). Quality-of-experience-oriented autonomous intersection control in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):1956–1967.
- [Das and Maurya, 2019] Das, S. and Maurya, A. K. (2019). Defining time-to-collision thresholds by the type of lead vehicle in non-lane-based traffic environments. *IEEE Transactions on Intelligent Transportation Systems*, 21(12):4972–4982.
- [Dayan and Hinton, 1993] Dayan, P. and Hinton, G. E. (1993). Feudal reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 5:271–278.
- [de Dios Ortúzar, 2021] de Dios Ortúzar, J. (2021). Future transportation: Sustainability, complexity and individualization of choices. *Communications in Transportation Research*.
- [Demir and Sezer, 2018] Demir, A. and Sezer, V. (2018). Intersection navigation under dynamic constraints using deep reinforcement learning. *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–5.
- [Deng, 2020] Deng, D. (2020). Dbscan clustering algorithm based on density. In *2020 7th international forum on electrical engineering and automation (IFEEA)*, pages 949–953. IEEE.

- [Dresner and Stone, 2004a] Dresner, K. and Stone, P. (2004a). Multiagent traffic management: A reservation-based intersection control mechanism. In *Autonomous Agents and Multiagent Systems, International Joint Conference on*, volume 3, pages 530–537. Citeseer.
- [Dresner and Stone, 2004b] Dresner, K. and Stone, P. (2004b). A reservation-based multiagent system for intersection control. *IFAC Proceedings Volumes*, 37(8):136–141.
- [Dresner and Stone, 2008] Dresner, K. and Stone, P. (2008). A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656.
- [Eiffert et al., 2020] Eiffert, S., Kong, H., Pirmarzashti, N., and Sukkarieh, S. (2020). Path planning in dynamic environments using generative rnns and monte carlo tree search. *arXiv preprint arXiv:2001.11597*.
- [Elhadef, 2015] Elhadef, M. (2015). An adaptable invanets-based intersection traffic control algorithm. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2387–2392.
- [Eom and Kim, 2020] Eom, M. and Kim, B.-I. (2020). The traffic signal control problem for intersections: a review. *European transport research review*, 12:1–20.
- [Fayazi and Vahidi, 2017] Fayazi, S. A. and Vahidi, A. (2017). Vehicle-in-the-loop (vil) verification of a smart city intersection control scheme for autonomous vehicles. *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1575–1580.
- [Federal Highway Administration, 2004] Federal Highway Administration (2004). Traffic control at signalized intersections.
<https://www.fhwa.dot.gov/publications/research/safety/04091/07.cfm>.
- [Federal Highway Administration, 2006] Federal Highway Administration (2006). Travel time reliability: Making it there on time, all the time. Technical report, U.S. Department of Transportation.
- [Feng et al., 2018] Feng, Y., Yu, C., and Liu, H. X. (2018). Spatiotemporal intersection control in a connected and automated vehicle environment. *Transportation Research Part C-emerging Technologies*, 89:364–383.
- [Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.
- [Fioretto et al., 2018] Fioretto, F., Pontelli, E., and Yeoh, W. (2018). Distributed constraint optimization problems and applications: A survey. *Journal of Artificial Intelligence Research*, 61:623–698.
- [Garey et al., 1976] Garey, M. R., Johnson, D. S., and Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2):117–129.

- [Garivier and Moulines, 2011] Garivier, A. and Moulines, E. (2011). On upper-confidence bound policies for switching bandit problems. In *International conference on algorithmic learning theory*, pages 174–188. Springer.
- [Ge et al., 2022] Ge, Q., Sartoretti, G., Duan, J., Li, S. E., Yin, Y., and Zheng, S. (2022). Distributed model predictive control of connected multi-vehicle systems at unsignalized intersections. *2022 IEEE International Conference on Unmanned Systems (ICUS)*, pages 1466–1472.
- [Gebauer et al., 2021] Gebauer, C., Dengler, N., and Bennewitz, M. (2021). Sensor-based navigation using hierarchical reinforcement learning. *arXiv preprint arXiv:2108.13268*.
- [Gregoire et al., 2014] Gregoire, J., Bonnabel, S., and de La Fortelle, A. (2014). Priority-based coordination of robots.
- [Gu et al., 2024] Gu, H., Wang, S., Ma, X., Jia, D., Mao, G., Lim, E. G., and Wong, C. P. R. (2024). Large-scale traffic signal control using constrained network partition and adaptive deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):7619–7632.
- [Guériau and Dusparic, 2020] Guériau, M. and Dusparic, I. (2020). Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8.
- [Gunarathna et al., 2022] Gunarathna, U., Karunasekera, S., Borovica-Gajic, R., and Tanin, E. (2022). Real-Time Intelligent Autonomous Intersection Management Using Reinforcement Learning. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 135–144. IEEE.
- [Guo et al., 2019] Guo, Y., Ma, J., Xiong, C., Li, X. S., Zhou, F., and Hao, W. (2019). Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation Research Part C: Emerging Technologies*.
- [Häfner et al., 2021] Häfner, B., Bajpai, V., Ott, J., and Schmitt, G. A. (2021). A survey on cooperative architectures and maneuvers for connected and automated vehicles. *IEEE Communications Surveys & Tutorials*, 24(1):380–403.
- [Hamouda et al., 2021] Hamouda, A. H., Mahfouz, D. M., Elias, C. M., and Shehata, O. M. (2021). Multi-layer Control Architecture for Unsignalized Intersection Management via Nonlinear MPC and Deep Reinforcement Learning. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1990–1996. IEEE.
- [Hausknecht et al., 2011] Hausknecht, M. J., Au, T.-C., and Stone, P. (2011). Autonomous intersection management: Multi-intersection optimization. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4581–4586.
- [He et al., 2024] He, J., Zhang, Y., and Zhou, Z. (2024). Autonomous-vehicle scheduling at signal-free intersections based on multi-agent deep reinforcement learning. In *Proceedings of SPIE*, volume 13018, page 1301853.

- [He et al., 2019a] He, R., Schneider, C., Ai, B., Wang, G., Zhong, Z., Dupleich, D. A., Thomae, R. S., Boban, M., Luo, J., and Zhang, Y. (2019a). Propagation channels of 5g millimeter-wave vehicle-to-vehicle communications: Recent advances and future challenges. *IEEE vehicular technology magazine*, 15(1):16–26.
- [He et al., 2019b] He, W., Li, H., Zhi, X., Li, X., Zhang, J., Hou, Q., and Li, Y. (2019b). Overview of v2v and v2i wireless communication for cooperative vehicle infrastructure systems. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 127–134.
- [Homayouni and Fontes, 2021] Homayouni, S. M. and Fontes, D. B. (2021). Production and transport scheduling in flexible job shop manufacturing systems. *Journal of Global Optimization*, 79(2):463–502.
- [Hu et al., 2018] Hu, J., Sun, Y., and Ban, X. (2018). Platoon-based intersection management in a connected and automated vehicle environment: A review. *Transportation Research Part C: Emerging Technologies*, 93:332–350.
- [Huang et al., 2023] Huang, S.-C., Lin, K.-E., Kuo, C.-Y., Lin, L.-H., Sayin, M. O., and Lin, C.-W. (2023). Reinforcement-learning-based job-shop scheduling for intelligent intersection management. *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6.
- [Hughes et al., 2010] Hughes, W., Jagannathan, R., Sengupta, D., Hummer, J., et al. (2010). Alternative intersections/interchanges: informational report (aiir). Technical report, United States. Federal Highway Administration. Office of Research
- [Hunt et al., 1982] Hunt, P. B., Robertson, D., Bretherton, R., and Winton, R. (1982). Scoot—a traffic responsive method of coordinating signals. *Transport and road research laboratory report*, 1014.
- [Iliopoulou et al., 2022] Iliopoulou, C. A., Kampitakis, E. P., Kepaptsoglou, K. L., and Vlahogianni, E. I. (2022). Dynamic traffic-aware auction-based signal control under vehicle to infrastructure communication. *Physica A: Statistical Mechanics and its Applications*.
- [Iqbal et al., 2021] Iqbal, S., Sha, F., and Lewis, M. (2021). Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Learning Representations*.
- [Iwase et al., 2022] Iwase, T., Stein, S., Gerding, E. H., and Chapman, A. (2022). A polynomial-time decentralised algorithm for coordinated management of multiple intersections. In Raedt, L. D., editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3847–3853. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- [Jin et al., 2013] Jin, Q., Wu, G., Boriboonsomsin, K., and Barth, M. (2013). Platoon-based multi-agent intersection management for connected vehicle. In *16th international ieee conference on intelligent transportation systems (itsc 2013)*, pages 1462–1467. IEEE.

- [Jing et al., 2017] Jing, P., Huang, H., and Chen, L. (2017). An adaptive traffic signal control in a connected vehicle environment: A systematic review. *Information*, 8(3):101.
- [Kai et al., 2020] Kai, S., Wang, B., Chen, D., Hao, J., Zhang, H., and Liu, W. (2020). A multi-task reinforcement learning approach for navigating unsignalized intersections. *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1583–1588.
- [Kamal et al., 2015] Kamal, M. A. S., Imura, J.-i., Hayakawa, T., Ohata, A., and Aihara, K. (2015). A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1136–1147.
- [Kamran et al., 2020] Kamran, D., Lopez, C. F., Lauer, M., and Stiller, C. (2020). Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1205–1212. IEEE.
- [Kaoud et al., 2017] Kaoud, E., El-Sharief, M. A., and El-Sebaie, M. (2017). Scheduling problems of automated guided vehicles in job shop, flow shop, and container terminals. In *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 60–65. IEEE.
- [Katole and Sinha, 2023] Katole, R. and Sinha, A. (2023). Autonomous intersection management for non-communicative autonomous vehicles. *arXiv preprint arXiv:2311.17681*.
- [Katriniok et al., 2017] Katriniok, A., Kleibaum, P., and Joševski, M. (2017). Distributed model predictive control for intersection automation using a parallelized optimization approach. *IFAC-PapersOnLine*, 50(1):5940–5946.
- [Katriniok et al., 2018] Katriniok, A., Kojchev, S., Lefeber, E., and Nijmeijer, H. (2018). Distributed scenario model predictive control for driver aided intersection crossing. In *2018 European Control Conference (ECC)*, page 1746–1752. IEEE.
- [Katriniok et al., 2019] Katriniok, A., Sopasakis, P., Schuurmans, M., and Patrinos, P. (2019). Nonlinear model predictive control for distributed motion planning in road intersections using panoc. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5272–5278. IEEE.
- [Keskin et al., 2020] Keskin, M. F., Peng, B., Kulcsár, B., and Wymeersch, H. (2020). Altruistic control of connected automated vehicles in mixed-autonomy multi-lane highway traffic. *IFAC-PapersOnLine*, 53:14966–14971.
- [Keyvan-Ekbatani et al., 2015] Keyvan-Ekbatani, M., Yildirimoglu, M., Geroliminis, N., and Papageorgiou, M. (2015). Multiple concentric gating traffic control in large-scale urban networks. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2141–2154.
- [Khayatian et al., 2019] Khayatian, M., Lou, Y., Mehrabian, M., and Shrivastava, A. (2019). Crossroads+. *ACM Transactions on Cyber-Physical Systems*, 4:1 – 28.
- [Khayatian et al., 2018a] Khayatian, M., Mehrabian, M., and Shrivastava, A. (2018a). Rim: Robust intersection management for connected autonomous vehicles. In *2018 IEEE Real-Time Systems Symposium (RTSS)*, page 35–44. IEEE.

- [Khayatian et al., 2018b] Khayatian, M., Mehrabian, M., and Shrivastava, A. (2018b). Rim: Robust intersection management for connected autonomous vehicles. *2018 IEEE Real-Time Systems Symposium (RTSS)*, pages 35–44.
- [Kim and Sung, 2023] Kim, W. and Sung, Y. (2023). Parameter sharing with network pruning for scalable multi-agent deep reinforcement learning. *arXiv preprint arXiv:2303.00912*.
- [Kumaravel et al., 2021] Kumaravel, S. D., Malikopoulos, A. A., and Ayyagari, R. (2021). Optimal coordination of platoons of connected and automated vehicles at signal-free intersections. *IEEE Transactions on Intelligent Vehicles*, 7(2):186–197.
- [LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- [Lee and Hess, 2020] Lee, D. and Hess, D. J. (2020). Regulations for on-road testing of connected and automated vehicles: Assessing the potential for global safety harmonization. *Transportation Research Part A: Policy and Practice*.
- [Li et al., 2021a] Li, B., Zhang, Y., Acarman, T., Ouyang, Y., Yaman, C., and Wang, Y. (2021a). Lane-free autonomous intersection management: A batch-processing framework integrating reservation-based and planning-based methods. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7915–7921.
- [Li et al., 2018] Li, B., Zhang, Y., Zhang, Y., Jia, N., and Ge, Y. (2018). Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, page 1432–1437. IEEE.
- [Li et al., 2022] Li, D., Wu, J., Zhu, F., Chen, T., and Wong, Y. D. (2022). Modeling adaptive platoon and reservation based autonomous intersection control: a deep reinforcement learning approach. *arXiv preprint arXiv:2206.12419*.
- [Li et al., 2023a] Li, D., Zhu, F., Chen, T., Wong, Y. D., Zhu, C., and Wu, J. (2023a). Coor-plt: A hierarchical control model for coordinating adaptive platoons of connected and autonomous vehicles at signal-free intersections based on deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 146:103933.
- [Li et al., 2023b] Li, J., Yin, B., and Fang, Y. (2023b). Collection auctions-based autonomous intersection management. *Machines*.
- [Li et al., 2024] Li, S., Peng, K., Hui, F., Li, Z., Wei, C., and Wang, W. (2024). A decision-making approach for complex unsignalized intersection by deep reinforcement learning. *IEEE Transactions on Vehicular Technology*.
- [Li et al., 2021b] Li, S., Wang, X., and Chen, G. (2021b). Group coordinated control of networked mobile robots with applications to autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4321–4332.

- [Li and Ibanez-Guzman, 2020] Li, Y. and Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61.
- [Li et al., 2019] Li, Z., Wu, Q., Yu, H., Chen, C., Zhang, G., Tian, Z. Z., and Prevedouros, P. D. (2019). Temporal-spatial dimension extension-based intersection control formulation for connected and autonomous vehicle systems. *Transportation Research Part C: Emerging Technologies*, 104:234–248.
- [Liang et al., 2018] Liang, X., Yan, T., Lee, J., and Wang, G. (2018). A distributed intersection management protocol for safety, efficiency, and driver’s comfort. *IEEE Internet of Things Journal*, 5:1924–1935.
- [Lillicrap et al., 2016] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*.
- [Lim et al., 2021] Lim, M. H., Tomlin, C. J., and Sunberg, Z. N. (2021). Voronoi progressive widening: efficient online solvers for continuous state, action, and observation pomdps. In *2021 60th IEEE conference on decision and control (CDC)*, pages 4493–4500. IEEE.
- [Lin et al., 2022] Lin, L.-H., Wang, K.-C., Lee, Y.-H., Lin, K.-E., Lin, C.-W., and Jiang, I. H.-R. (2022). Deadlock resolution for intelligent intersection management with changeable trajectories. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 573–579. IEEE.
- [Liu et al., 2023] Liu, J., Hang, P., Na, X., Huang, C., and Sun, J. (2023). Cooperative decision-making for cavs at unsignalized intersections: A marl approach with attention and hierarchical game priors. *Authorea Preprints*.
- [Liu et al., 2025] Liu, S., Fan, M., Li, W., Du, J., and Li, S. (2025). Large-scale mixed-traffic and intersection control using multi-agent reinforcement learning. *arXiv preprint arXiv:2504.04691*.
- [Liu and et al., 2020] Liu, Y. and et al. (2020). Edge computing for autonomous driving: Opportunities and challenges. *IEEE Network*, 34(3):3–9.
- [Lopez et al., 2018a] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wiessner, E. (2018a). Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582.
- [Lopez et al., 2018b] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Krajzewicz, D., Wagner, P., and Wießner, E. (2018b). Sumo—simulation of urban mobility: An overview. In *Proceedings of the International Conference on Advances in System Simulation (SIMUL)*, pages 55–60.
- [Lou et al., 2022] Lou, Y., Wu, J., and Ran, Y. (2022). Meta-reinforcement learning for multiple traffic signals control. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4264–4268.

- [Lowe et al., 2017] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*.
- [Lu and Kim, 2016] Lu, Q. and Kim, K.-D. (2016). Intelligent intersection management of autonomous traffic using discrete-time occupancies trajectory. *Journal of Traffic and Logistics Engineering*.
- [Ma et al., 2019] Ma, X., Driggs-Campbell, K., Zhang, Z., and Kochenderfer, M. J. (2019). Monte carlo tree search for policy optimization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3634–3640.
- [Mahbub, 2020] Mahbub, A M Ishtiaque; Malikopoulos, A. A. Z. L. (2020). Decentralized optimal coordination of connected and automated vehicles for multiple traffic scenarios. *Automatica*, 117(117):108958–NA.
- [Mirheli et al., 2019] Mirheli, A., Tajalli, M., Hajibabai, L., and Hajbabaie, A. (2019). A consensus-based distributed trajectory control in a signal-free intersection. *Transportation Research Part C: Emerging Technologies*, 100:161–176.
- [Mitrovic et al., 2020] Mitrovic, N., Dakic, I., and Stevanovic, A. Z. (2020). Combined alternate-direction lane assignment and reservation-based intersection control. *IEEE Transactions on Intelligent Transportation Systems*, 21:1779–1789.
- [Mnih et al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- [Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.
- [Mohamed et al., 2021] Mohamed, W., Zhang, H., and Wang, X. (2021). Monte carlo tree search: A survey of recent modifications and applications. *Artificial Intelligence Review*, 54(6):4819–4865.
- [Morrissett et al., 2022] Morrissett, A., Martin, P. J., and Abdelwahed, S. (2022). Socially-optimal auction-theoretic intersection management system. *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1340–1346.
- [Mushtaq et al., 2021] Mushtaq, A., ul Haq, I., un Nabi, W., Khan, A., and Shafiq, O. (2021). Traffic flow management of autonomous vehicles using platooning and collision avoidance strategies. *Electronics*.
- [Muzzini and Montangero, 2024] Muzzini, F. and Montangero, M. (2024). Exploiting traffic light coordination and auctions for intersection and emergency vehicle management in a smart city mixed scenario. *Sensors*, 24(7):2036.

- [Nachum et al., 2018] Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3303–3313.
- [Narvekar et al., 2020] Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50.
- [Naveed et al., 2021] Naveed, K. B., Qiao, Z., and Dolan, J. M. (2021). Trajectory Planning for Autonomous Vehicles Using Hierarchical Reinforcement Learning. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 601–606. IEEE.
- [Noaeen et al., 2022] Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z. S. H., Bazzan, A. L., and Far, B. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, 199:116830.
- [Paden et al., 2016] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55.
- [Pairet et al., 2022] Pairet, , Petillot, Y., and Lane, D. (2022). Monte-carlo robot path planning. *IEEE Robotics and Automation Letters*, 7(2):3001–3008.
- [Pan et al., 2022] Pan, A., Bhatia, K., and Steinhardt, J. (2022). The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*.
- [Pan et al., 2024] Pan, Y., Wu, Y., Xu, L., Xia, C., and Olson, D. L. (2024). The impacts of connected autonomous vehicles on mixed traffic flow: A comprehensive review. *Physica A: Statistical Mechanics and its Applications*, page 129454.
- [Papoudakis et al., 2021] Papoudakis, G., Christianos, F., Schäfer, L., and Albrecht, S. (2021). Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *Proceedings of the Neural Information Processing Systems Conference*.
- [Papoudakis et al., 2019] Papoudakis, G., Christianos, F., Schäfer, L., and Albrecht, S. V. (2019). Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*.
- [Pathak et al., 2017] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2778–2787. PMLR.
- [Pei et al., 2021] Pei, H., Zhang, Y., Tao, Q., Feng, S., and Li, L. (2021). Distributed cooperative driving in multi-intersection road networks. *IEEE Transactions on Vehicular Technology*, 70:5390–5403.
- [Pendyala et al., 2024] Pendyala, A., Atamna, A., and Glasmachers, T. (2024). Solving a real-world optimization problem using proximal policy optimization with curriculum learning and reward

- engineering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 150–165. Springer.
- [Peng et al., 2023] Peng, X., Gao, H., Han, G., Wang, H., and Zhang, M. (2023). Joint optimization of traffic signal control and vehicle routing in signalized road networks using multi-agent deep reinforcement learning. *arXiv preprint arXiv:2310.10856*.
- [Pires et al., 2023] Pires, B. A., Behbahani, F., Soyer, H., Nikiforou, K., Keck, T., and Singh, S. (2023). Hierarchical reinforcement learning in complex 3d environments. *arXiv preprint arXiv:2302.14451*.
- [Puterman, 1994] Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
- [Qi et al., 2022] Qi, R., Huang, J., Li, H., Tan, Q., Huang, L., and Cui, J. (2022). Random ensemble reinforcement learning for traffic signal control. *arXiv preprint arXiv:2203.05961*.
- [Qiao et al., 2018a] Qiao, J., Zhang, D., and de Jonge, D. (2018a). Virtual roundabout protocol for autonomous vehicles. In *Australasian Conference on Artificial Intelligence*.
- [Qiao et al., 2018b] Qiao, Z., Muelling, K., Dolan, J. M., Palanisamy, P., and Mudalige, P. (2018b). Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1233–1238. IEEE.
- [Qiao et al., 2021] Qiao, Z., Schneider, J., and Dolan, J. M. (2021). Behavior Planning at Urban Intersections through Hierarchical Reinforcement Learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2667–2673. IEEE.
- [Rasheed et al., 2020] Rasheed, F., Yau, K.-L. A., Noor, R. M., Wu, C., and Low, Y.-C. (2020). Deep reinforcement learning for traffic signal control: A review. *IEEE Access*, 8:208016–208044.
- [Rios-Torres and Malikopoulos, 2016] Rios-Torres, J. and Malikopoulos, A. A. (2016). A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077.
- [Ruan et al., 2024] Ruan, J., Li, Z., Wei, H., Jiang, H., Lu, J., Xiong, X., Mao, H., and Zhao, R. (2024). Coslight: Co-optimizing collaborator selection and decision-making to enhance traffic signal control. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2500–2511.
- [SAE International, 2018] SAE International (2018). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE Standard J3016_2018. Accessed on January 13, 2025.
- [Santos et al., 2020] Santos, J.-A., Fernández, A., Moreno-Rebato, M., Billhardt, H., Rodríguez-García, J.-A., and Ossowski, S. (2020). Legal and ethical implications of applications based on agreement technologies: the case of auction-based road intersections. *Artificial Intelligence and Law*, 28(4):385–414.

- [Sarker, 2023] Sarker, S. (2023). Analyzing behaviors of mixed traffic via reinforcement learning at unsignalized intersections.
- [Schmidt et al., 2024] Schmidt, J., Dreyer, F., Hashimi, S. A., and Stober, S. (2024). Transferlight: Zero-shot traffic signal control on any road-network. *Preprint*. ArXiv preprint.
- [Schulman et al., 2015] Schulman, J., Moritz, P., Levine, S., Jordan, M. I., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.
- [Schulman et al., 2017a] Schulman, J., Wolski, F., Dhariwal, P., et al. (2017a). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- [Schulman et al., 2017b] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017b). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. Available at: <https://arxiv.org/abs/1707.06347>.
- [Seong et al., 2021] Seong, H., Jung, C., Lee, S., and Shim, D. H. (2021). Learning to drive at unsignalized intersections using attention-based deep reinforcement learning. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 559–566.
- [Serafini and Ukovich, 1989] Serafini, P. and Ukovich, W. (1989). A mathematical model for the fixed-time traffic control problem. *European Journal of Operational Research*, 42(2):152–165.
- [Shao et al., 2024] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- [Shi et al., 2024a] Shi, Y., Wang, W., Tao, X., Dusparic, I., and Cahill, V. (2024a). Applying neural monte carlo tree search to unsignalized multi-intersection scheduling for autonomous vehicles. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 935–942. IEEE.
- [Shi et al., 2024b] Shi, Y., Wang, W., Tao, X., Dusparic, I., and Cahill, V. (2024b). Applying neural monte carlo tree search to unsignalized multi-intersection scheduling for autonomous vehicles. *arXiv preprint arXiv:2410.18786*.
- [Shikunov and Panov, 2020] Shikunov, M. and Panov, A. I. (2020). Hierarchical reinforcement learning approach for the road intersection task. In *Biologically Inspired Cognitive Architectures 2019: Proceedings of the Tenth Annual Meeting of the BICA Society 10*, pages 495–506. Springer.
- [Shinde, 2017] Shinde, S. M. (2017). Adaptive traffic light control system. In *2017 1st international conference on intelligent systems and information management (ICISIM)*, pages 300–306. IEEE.
- [Silver et al., 2018] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

- [Silver et al., 2014] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr.
- [Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.
- [Sims and Dobinson, 1980] Sims, A. G. and Dobinson, K. W. (1980). The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on vehicular technology*, 29(2):130–137.
- [Soydaner, 2020] Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13):2052013.
- [Spatharis and Blekas, 2024] Spatharis, C. and Blekas, K. (2024). Multiagent reinforcement learning for autonomous driving in traffic zones with unsignalized intersections. *Journal of Intelligent Transportation Systems*, 28(1):103–119.
- [Stevanovic et al., 2010] Stevanovic, A., Martin, P. T., and Stevanovic, J. (2010). Adaptive traffic control systems: Domestic and foreign state of practice. *Transportation Research Board*, 403.
- [Su et al., 2020] Su, S., Chaniotakis, E., Narayanan, S., Jiang, H., and Antoniou, C. (2020). Clustered tabu search optimization for reservation-based shared autonomous vehicles. *Transportation Letters*, 14:124 – 128.
- [Su et al., 2023] Su, Z., Chow, A. H., Fang, C., Liang, E., and Zhong, R. (2023). Hierarchical control for stochastic network traffic with reinforcement learning. *Transportation Research Part B: Methodological*, 167:196–216.
- [Sun et al., 2023] Sun, Q., Zhang, L., Yu, H., Zhang, W., Mei, Y., and Xiong, H. (2023). Hierarchical reinforcement learning for dynamic autonomous vehicle navigation at intelligent intersections. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4852–4861.
- [Sunberg and Kochenderfer, 2018] Sunberg, Z. and Kochenderfer, M. (2018). Online algorithms for pomdps with continuous state, action, and observation spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, pages 259–263.
- [Suriyarachchi and Baras, 2021] Suriyarachchi, N. and Baras, J. S. (2021). Shock wave mitigation in multi-lane highways using vehicle-to-vehicle communication. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 1–7. IEEE.
- [Sutton, 2018] Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.
- [Sutton et al., 1999a] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999a). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1057–1063. MIT Press.

- [Sutton et al., 1999b] Sutton, R. S., Precup, D., and Singh, S. (1999b). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.
- [Świechowski et al., 2023] Świechowski, M., Godlewski, K., Sawicki, B., and Mańdziuk, J. (2023). Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562.
- [Tachet et al., 2016] Tachet, R., Santi, P., Sobolevsky, S., Reyes-Castro, L. I., Frazzoli, E., Helbing, D., and Ratti, C. (2016). Revisiting street intersections using slot-based systems. *PLoS ONE*, 11.
- [taher et al., 2020] taher, f., Elmahalawy, A., Shouman, A., and EL-SAYED, A. (2020). Applying Hierarchal Clusters on Deep Reinforcement Learning Controlled Traffic Network. *Menoufia Journal of Electronic Engineering Research*, 0(0):0–0.
- [Talebpour and Mahmassani, 2016] Talebpour, A. and Mahmassani, H. S. (2016). Influence of autonomous and connected vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143–163.
- [Tallapragada et al., 2023] Tallapragada, P. et al. (2023). Reinforcement learning aided sequential optimization for unsignalized intersection management of robot traffic. *arXiv preprint arXiv:2302.05082*.
- [Tan et al., 2020] Tan, T., Bao, F., Deng, Y., Jin, A., Dai, Q., and Wang, J. (2020). Cooperative Deep Reinforcement Learning for Large-Scale Traffic Grid Signal Control. *IEEE Transactions on Cybernetics*, 50(6):2687–2700.
- [Tang et al., 2019] Tang, K., Boltze, M., Nakamura, H., and Tian, Z. (2019). *Global practices on road traffic signal control: Fixed-time control at isolated intersections*. Elsevier.
- [Tatarek et al., 2017] Tatarek, T., Kronenberger, J., and Handmann, U. (2017). Functionality, advantages and limits of the tesla autopilot.
- [Texas Transportation Institute, 2023] Texas Transportation Institute (2023). Urban mobility report. <https://mobility.tamu.edu/umr/report/>.
- [Tlig et al., 2014] Tlig, M., Buffet, O., and Simonin, O. (2014). Stop-free strategies for traffic networks: Decentralized on-line optimization. In *European Conference on Artificial Intelligence*.
- [Treiber and Kesting, 2013] Treiber, M. and Kesting, A. (2013). *Traffic Flow Dynamics: Data, Models and Simulation*. Springer.
- [U.K. Government Service, 2016] U.K. Government Service (2016). Transport systems catapult: The pathway to cav deployment. Accessed on January 13, 2025.
- [Van Hasselt et al., 2016] Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

- [Vasirani and Ossowski, 2011] Vasirani, M. and Ossowski, S. (2011). A computational market for distributed control of urban road traffic systems. *IEEE Transactions on Intelligent Transportation Systems*, 12:313–321.
- [Vasirani and Ossowski, 2014] Vasirani, M. and Ossowski, S. (2014). A market-inspired approach for intersection management in urban road traffic networks. *J. Artif. Intell. Res.*, 43:621–659.
- [Vu et al., 2018] Vu, H., Aknine, S., and Ramchurn, S. D. (2018). A decentralised approach to intersection traffic management. In *IJCAI*, pages 527–533.
- [Wang et al., 2020a] Wang, F., Shi, D., Liu, T., and Tang, X. (2020a). Decision-making at unsignalized intersection for autonomous vehicles: Left-turn maneuver with deep reinforcement learning. *arXiv preprint arXiv:2008.06595*.
- [Wang et al., 2018] Wang, J., Zhao, X., and Yin, G. (2018). Multiobjective optimal cooperative driving for connected and automated vehicles at nonsignalised intersection. *IET Intelligent Transport Systems*, 13(1):79–89.
- [Wang et al., 2019] Wang, S., Liu, H., Niyato, D., Zhao, Y., and Han, Z. (2019). Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. In *2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- [Wang et al., 2020b] Wang, Y., Cai, P., and Lu, G. (2020b). Cooperative autonomous traffic organization method for connected automated vehicles in multi-intersection road networks. *Transportation research part C: emerging technologies*, 111:458–476.
- [Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.
- [Webster, 1958] Webster, F. V. (1958). Traffic signal settings.
- [Wei et al., 2019a] Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., and Li, Z. (2019a). Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1290–1298.
- [Wei et al., 2019b] Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., and Li, Z. (2019b). Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1290–1298.
- [Wei et al., 2019c] Wei, H., Chen, G., Zheng, G., and Wu, Z. (2019c). Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [Wei et al., 2019d] Wei, H., Xu, N., Zhang, H., and Li, Z. (2019d). Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922. ACM.

- [Weingertner et al., 2020] Weingertner, P., Ho, M., Timofeev, A., Aubert, S., and Gil, G. P. (2020). Monte carlo tree search with reinforcement learning for motion planning. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7.
- [Wiering, 2000] Wiering, M. (2000). Multi-agent reinforcement learning for traffic light control. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 1151–1158.
- [Wu et al., 2021a] Wu, F., Yin, Y., Lawphongpanich, S., and Yang, H. (2021a). Real-time traffic signal control optimization with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*.
- [Wu and Qu, 2022] Wu, J. and Qu, X. (2022). Intersection control with connected and automated vehicles: A review. *Journal of intelligent and connected vehicles*, 5(3):260–269.
- [Wu et al., 2021b] Wu, Q., Zhang, L., Shen, J., Lü, L., Du, B., and Wu, J. (2021b). Efficient pressure: Improving efficiency for signalized intersections. *arXiv preprint arXiv:2112.02336*.
- [Wu et al., 2020] Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., and Wu, D. O. (2020). Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(8):8243–8256.
- [Wu et al., 2019a] Wu, W., Liu, Y., Guo, Z., Ni, C., and Qiao, Y. (2019a). Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133. Accessed: February 23, 2025.
- [Wu et al., 2022] Wu, W., Liu, Y., Liu, W., Zhang, F., Dixit, V. V., and Waller, S. T. (2022). Autonomous intersection management for connected and automated vehicles: A lane-based method. *IEEE Transactions on Intelligent Transportation Systems*, 23:15091–15106.
- [Wu, 2022] Wu, Jiaming; Qu, X. (2022). Intersection control with connected and automated vehicles: a review. *Journal of Intelligent and Connected Vehicles*, 5(3):260–269.
- [Wu et al., 2019b] Wu, Y., Chen, H., and Zhu, F. (2019b). Dcl-aim: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 103:246–260.
- [Xu et al., 2021a] Xu, B., Wang, Y., Wang, Z., Jia, H., and Lu, Z. (2021a). Hierarchically and cooperatively learning traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 669–677.
- [Xu et al., 2020] Xu, H., Zhang, Y., Li, L., and Li, W. (2020). Cooperative driving at unsignalized intersections using tree search. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4563–4571.
- [Xu, 2018] Xu, Biao; Li, S. E. Y. B. L. S. B. X. J. W. J. L. K. (2018). Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections. *Transportation Research Part C: Emerging Technologies*, 93(NA):322–334.

- [Xu et al., 2021b] Xu, Y., Zhou, H., Ma, T., Zhao, J., Qian, B., and Shen, X. (2021b). Leveraging multiagent learning for automated vehicles scheduling at nonsignalized intersections. *IEEE Internet of Things Journal*, 8(14):11427–11439.
- [Xue et al., 2022] Xue, W., Wu, H., Ye, H., and Shao, S. (2022). An improved proximal policy optimization method for low-level control of a quadrotor. In *Actuators*, volume 11, page 105. MDPI.
- [Yang et al., 2021] Yang, D., Zhang, H., Lan, X., and Ding, J. (2021). Density-based curriculum for multi-goal reinforcement learning with sparse rewards. *arXiv preprint arXiv:2109.08903*.
- [Yao, 2020] Yao, Handong; Li, X. (2020). Decentralized control of connected automated vehicle trajectories in mixed traffic at an isolated signalized intersection. *Transportation Research Part C: Emerging Technologies*, 121(NA):102846–NA.
- [Yau et al., 2017] Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., and Komisarczuk, P. (2017). A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys (CSUR)*, 50(3):1–38.
- [Yeong et al., 2021] Yeong, D. J., Velasco-Hernandez, G., Barry, J., and Walsh, J. (2021). Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140.
- [Yi et al., 2022] Yi, Y., Li, G., Wang, Y., and Lu, Z. (2022). Learning to share in networked multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:15119–15131.
- [Yildirimoglu et al., 2018] Yildirimoglu, M., Sirmatel, I. I., and Geroliminis, N. (2018). Hierarchical control of heterogeneous large-scale urban road networks via path assignment and regional route guidance. *Transportation Research Part B: Methodological*, 118:106–123.
- [Yu et al., 2019] Yu, C., Sun, W., Liu, H. X., and Yang, X. (2019). Managing connected and automated vehicles at isolated intersections: From reservation-to optimization-based methods. *Transportation research part B: methodological*, 122:416–435.
- [Yu et al., 2022] Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624.
- [Zang et al., 2020a] Zang, H., Hu, Y., Zhang, Y., Guo, S. E. L., and Song, L. (2020a). Mplight: Efficient reinforcement learning for urban traffic signal control. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [Zang et al., 2020b] Zang, X., Yao, H., Zheng, G., and Li, Z. (2020b). Metalight: Value-based meta-reinforcement learning for online universal traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages XXX–XXX.
- [Zang et al., 2020c] Zang, X., Yao, H., Zheng, G., Xu, N., Xu, K., and Li, Z. (2020c). Metalight: Value-based meta-reinforcement learning for traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(1):1153–1160.

- [Zeng et al., 2022] Zeng, J., Xin, J., Cong, Y., Zhu, J., Zhang, Y., Jiang, W., and Pu, S. (2022). Halight: Hierarchical Deep Reinforcement Learning for Cooperative Arterial Traffic Signal Control with Cycle Strategy. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 479–485. IEEE.
- [Zhang et al., 2020a] Zhang, K., Zhang, L., and Xie, L. (2020a). *Discrete-time and discrete-space dynamical systems*. Springer.
- [Zhang et al., 2022] Zhang, L., Wu, Q., Shen, J., Lü, L., Du, B., and Wu, J. (2022). Expression might be enough: Representing pressure and demand for reinforcement learning based traffic signal control. In *International Conference on Machine Learning*, pages 26645–26654. PMLR.
- [Zhang and Lesser, 2020] Zhang, W. and Lesser, V. (2020). Monte-carlo tree search for scalable coalition formation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 409–415.
- [Zhang and Zhou, 2018] Zhang, Y. and Zhou, Y. (2018). Distributed coordination control of traffic network flow using adaptive genetic algorithm based on cloud computing. *J. Netw. Comput. Appl.*, 119:110–120.
- [Zhang et al., 2016] Zhang, Y. J., Malikopoulos, A. A., and Cassandras, C. G. (2016). Optimal control and coordination of connected and automated vehicles at urban traffic intersections. In *2016 American Control Conference (ACC)*, page 6227–6232. IEEE.
- [Zhang et al., 2020b] Zhang, Z., Liu, J., Zhang, H., et al. (2020b). Multi-agent deep q-network for unsignalized intersection management in a 2x2 grid. *IEEE Transactions on Intelligent Transportation Systems*.
- [Zhao et al., 2020] Zhao, C., Hu, X., and Wang, G. (2020). PDLight: A deep reinforcement learning traffic light control algorithm with pressure and dynamic light duration. *arXiv preprint arXiv:2009.13711*.
- [Zhao et al., 2023] Zhao, Y., Yan, X., Wen, S., Li, M., and Sun, H. (2023). A Driver-Friendly Cooperative Control Method for Connected and Automated Vehicles at An Unsignalized Intersection. In *2023 6th International Symposium on Autonomous Systems (ISAS)*, pages 1–6. IEEE.
- [Zheng et al., 2022] Zheng, J., Zhu, K., and Wang, R. (2022). Deep reinforcement learning for autonomous vehicles collaboration at unsignalized intersections. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 1115–1120.
- [Zhong et al., 2020] Zhong, Z., Nejad, M., and Lee, E. E. (2020). Autonomous and semi-autonomous intersection management: A survey. *IEEE Intelligent Transportation Systems Magazine*.
- [Zhou et al., 2022] Zhou, A., Peeta, S., Yang, M., and Wang, J. (2022). Cooperative signal-free intersection control using virtual platooning and traffic flow regulation. *Transportation research part C: emerging technologies*, 138:103610.

- [Zhou et al., 2020] Zhou, M., Liu, Z., Sui, P., Li, Y., and Chung, Y. Y. (2020). Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:11853–11864.
- [Zhou, 2022] Zhou, Anye; Peeta, S. Y. M. W. J. (2022). Cooperative signal-free intersection control using virtual platooning and traffic flow regulation. *Transportation Research Part C: Emerging Technologies*, 138(NA):103610–103610.