Imperial College London
Department of Civil and Environmental Engineering

Collaborative and Safe Autonomous Driving through Multi-Agent Deep Reinforcement Learning

Leandro Andres Parada Pradenas

Submitted in part fulfilment of the requirements
for the degree of Doctor of Philosophy
September 2024

I hereby declare that this thesis and the work reported herein was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the list of sources.

Leandro Parada

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International Licence (CC BY-NC-ND). Under this licence, you may copy and redistribute the material in any medium or format on the condition that: you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Autonomous Vehicles (AVs) have the potential to drastically improve traffic safety and efficiency by eliminating human error. Despite substantial advancements, fully autonomous deployment faces challenges in navigating unpredictable urban environments, ensuring cybersecurity, and coordinating multiple AVs in real-time. Traditional Reinforcement Learning methods succeed in isolated settings but fall short in multi-agent scenarios. This thesis uses Multi-Agent Reinforcement Learning (MARL) to improve AV decision-making in dynamic, real-world conditions.

This research develops secure MARL solutions for AV deployment, addressing coordination, communication, security, scalability, and sim-to-real challenges. Key contributions include a domain randomisation technique developed to enhance the transferability of simulation-trained MARL policies to real-world applications. A novel method for AV coordination in emergency scenarios using MARL together with risk-based modelling. Advanced communication strategies facilitate AV navigation in complex urban environments, particularly in occluded scenarios. A robust defence framework to protect against adversarial communication attacks, ensuring reliable network cooperation. Finally, a curriculum-based MARL approach is proposed for large-scale autonomous systems, demonstrating learning stability and efficiency.

Experimental results demonstrate the effectiveness of these contributions. The AV coordination method for emergency vehicles reduces collision risks and enables 15% higher speeds while maintaining high safety distances. Communication-based policies with LiDAR feature sharing reduce crash rates by 70%, and a communication scheduler decreases communication efforts by 60%. The defence mechanism shows zero-shot resilience in most adversarial scenarios, suffering 50% less disruption and quickly restoring cooperative performance. The scalable method for large-scale systems improves total reward by 31% while accelerating learning speed by 47%. Transferring trained policies with domain randomisation reduces the reality gap by 90%.

These advancements collectively enhance AV capabilities in coordination, scalability, security, and practical applicability. However, challenges remain in applying these solutions in real-world scenarios, such as addressing communication constraints, navigating regulatory and privacy concerns, and further bridging the reality gap for real-world scenarios.

Acknowledgements

The completion of my PhD study would not have been possible without the support I received during the past four years from many sides, and I would like to express my most sincere thanks:

- To my supervisor, Prof. Panagiotis Angeloudis, for his unwavering support throughout my PhD journey. His encouragement to continually push the boundaries of my research, combined with his always friendly and approachable nature, has been invaluable to my academic and personal growth.
- To all the members of the Transport Systems and Logistics Laboratory (TSL), with whom I had the opportunity to collaborate and engage in deep research discussions. In particular, I would like to thank Dr Eduardo Candela, Dr Felix Feng, Dr He-in Cheong, Dr Jose Escribano, Kenta Matsui, Luís Marques, Josh Kirk, Fahmy Adan, Hanlin Tian, Kevin Yu, Vincent Wu, Keyang Zhang for their inspiring conversations and valuable insights.
- To the Chilean Agency of Research and Development (ANID) for funding my PhD studies and providing the essential financial support that made this research possible.
- To my friends at the Civil and Environmental Engineering Department, who provided constant support and made this journey incredibly enjoyable. A special mention goes to Petrina, Karim and Meth, with whom I had the absolute pleasure of sharing this journey, forging friendships that will last a lifetime.
- To my Chilean friends in London, Nicolás, Giuliano, Ignacio, and Rodrigo, whom I am proud to call part of my family.
- To my family in Chile, who have loved and supported me at every step of my life.
- To Sonia, my love, for being the source of my greatest joy, my dearest friend, and for filling my life with the warmth of your love.

Contents

Abstract **3**

Acknowledgements **5**

1 Introduction **15**

1.1 Motivation 15

1.2 Decision-Making Framework for AVs 17

1.3 Research Aim and Objectives 19

1.4 Thesis Structure 20

1.5 Publications related to this work 21

2 Mathematical Foundations of Multi-Agent Reinforcement Learning **24**

2.1 Introduction 24

2.2 Markov Decision Process 25

2.3 Stochastic Games 28

2.4 Reinforcement Learning 31

2.5 Concluding Remarks 38

3 Literature Review **39**

3.1 Introduction 39

3.2 Autonomous Vehicle Systems 40

3.3 Multi-agent Communication and Adversarial Attacks 52

3.4 Autonomous Fleet Management 58

3.5 Sim-to-real Transfer for Autonomous Driving 62

3.6 Summary of gaps in the current literature 65

4 Ensuring Coordination and Safety in Real-World MARL through Sim-to-Real 67

4.1 Introduction 67

4.2 Framework 69

4.3 Experiments 74

4.4 Concluding Remarks 79

5 Coordinated Manoeuvring in Presence of Emergency Vehicles 81

5.1 Introduction 81

5.2 Modelling risk 83

5.3 Learning approach 87

5.4 Analysis 94

5.5 Concluding Remarks 108

6 Communication Strategies for Coordinated Autonomous Driving 111

6.1 Introduction 111

6.2 Improving Safety by Communicating Vehicle Intent 113

6.3 Safe Navigation of Occluded Intersections 125

6.4 Concluding Remarks 135

7 Robust Defence Against V2X Communication Attacks 137

7.1 Introduction 137

7.2 Problem Formulation 140

7.3 Framework 142

7.4 Experiments 147

7.5 Concluding Remarks 154

8 Scaling DRL For Autonomous Fleet Management 156

8.1 Introduction 156

8.2 Problem Statement . . . 159
8.3 DRL Architectures . . . 161
8.4 Curriculum-based Parameter Sharing DQN . . . 163
8.5 Experiments . . . 166
8.6 Concluding Remarks . . . 179

9 Conclusions and Future Work **181**
9.1 Summary of Key Findings and Contributions . . . 182
9.2 Research Impact and Practical Applications . . . 183
9.3 Challenges and Future Work . . . 185

A Implementation Details **189**
A.1 Coordination in the presence of EMVs (Chapter 5) . . . 189
A.2 TrustCommNet (Chapter 7) . . . 191
A.3 Curriculum-Based DQN (Chapter 8) . . . 198

B Popular MARL algorithms **201**
B.1 Deep Q-Network . . . 201
B.2 QMIX . . . 202
B.3 MAPPO . . . 203

C V2X Techonology and Standards **206**
C.1 V2X Technology . . . 206
C.2 Current V2X Standards . . . 207

Bibliography **209**

List of Tables

1.1 Thesis methodological chapters, decision-making levels, and challenges. 21

3.1 Summary of Literature on Autonomous Intersection Management Methods . . . 44

3.2 Summary of literature on the passing of EMVs in traffic. 45

3.3 Summary of the main gym environments for autonomous driving simulation. . . 53

3.4 Different DRL modelling approaches for dynamic fleet control. 63

4.1 Training Parameters for MARL Experiments 75

4.2 Domain randomisation distributions. 76

5.1 Comparison to the car following baselines. 99

5.2 Results of different reward functions. 102

5.3 Results of the experiments for different mixed scenarios. 103

5.4 Results for different number of agents. 106

5.5 Results of the experiments for different levels of collaboration. 107

6.1 Test arrival rate (%) of different Communicative MARL methods. 123

6.2 Test collision rates (%) for IntNet and IC3Net across various agent counts. . . . 124

6.3 Effect of the communication scheduler on graph density and success rate. 125

6.4 Result comparison for the occluded scenarios. 132

7.1 Performance degradation of MARL algorithms under adversarial communication. 149

7.2 Test performance after adversarial training 150

7.3 Zero-shot performance for the highway-env environment. 151

7.4 Scalability of zero-shot algorithm defence across different numbers of attackers. . 153

7.5 Ablation study: Impact of the Communication Scheduler on graph density and success rates. 154

8.1 Performance comparison of the different methods for the GridWorld case study. 171

8.2 Performance comparison of the different methods for the Manhattan case study. 177

A.1 Environment parameters. 191

A.2 Risk indices parameters 191

A.3 MAPPO parameters. 192

A.4 Hyper-parameters used to train TrustCommNet 194

A.5 Adversarial Training Times 194

A.6 DQN parameters. 199

A.7 Hyperparameters for PSDQN-CL 200

List of Figures

1.1 Decision-making levels in Autonomous Driving 18

3.1 Schematic overview of the MARL approach for Autonomous Driving 51
3.2 Visual representation of different autonomous driving gym environments 52
3.3 Challenging occluded scenario in an intersection. 55
3.4 Sim-to-real transfer for Autonomous Driving with human guidance. 64

4.1 Sim-to-real MARL at the operational level 68
4.2 Sim2Real setup. 69
4.3 Test track for the Sim2Real experiments. 72
4.4 Duckie-MAAD architecture. 73
4.5 Comparison of reward convergence with and without domain randomisation (D.R.) during training. 77
4.6 Average rewards for policies in simulation and real world. 77
4.7 Distribution of speed and off-track incidences across different policies. 78
4.8 Distribution of collisions and lane changes across different policies. 79

5.1 Manoeuvring around EMVs as a tactical decision-making level 82
5.2 Diagram of the risk-aware learning methodology 89
5.3 Graphic representation of the multi-agent autonomous traffic approach. 91
5.4 Representation of the multi-agent environment. 92
5.5 Comparison of the policies obtained with the three different methods. 100
5.6 Example of preemptive lane changing maneuvers by the AVs. 101
5.7 Speed Profiles for MPC and MAPPO methods 101
5.8 Influence of the risk component over the EMV speed and the risk of collision. . . 103

5.9 Speed Profiles for MAPPO-Controlled AVs with HDVs . . . 104
5.10 Learning curves for different number of agents. . . . 105
5.11 Cooperative vs. competitive multi-agent policies. . . . 108
5.12 Comparison between the cooperative and competitive scenarios . . . 109
5.13 Speed Profiles in Fully Cooperative vs Fully Competitive Scenarios. . . . 109

6.1 Communication at the tactical decision-making level. . . . 112
6.2 Illustration of our intention-sharing framework in an urban traffic context. . . . 115
6.3 IntNet framework for safe, efficient CAV navigation. . . . 116
6.4 Scenarios used for the experiments. . . . 120
6.5 Team reward of the different methods during training. . . . 120
6.6 Success rate of the different methods during training. . . . 121
6.7 Evolution of statistics for IntNet during training on the Highway scenario. . . . 123
6.8 Scalability analysis of the proposed method. . . . 124
6.9 Heatmaps of communication graphs. . . . 125
6.10 Collaborative CAV control. . . . 126
6.11 Collaborative MARL framework. . . . 129
6.12 Occluded scenarios used in the experiments. . . . 131
6.13 Learning curves for MAPPO and MAPPO with ground truth. . . . 133
6.14 Example trajectory obtained using the proposed method. . . . 134
6.15 Collision and Success rate as a function of LiDAR point dropout. . . . 135

7.1 Robust defence strategies against V2X communication attacks at the tactical decision-making level. . . . 138
7.2 Conceptual overview of TrustCommNet. . . . 139
7.3 TrustCommNet architecture diagram. . . . 143
7.4 Trust-based mechanism. . . . 145
7.5 Communication attacks on state-of-the-art MARL methods. . . . 148
7.6 Adversarial training performance comparison. . . . 150

7.7 Performance comparison of adversarial training between TrustCommNet and baseline models. . . . 151
7.8 TrustCommNet in action during a test episode. . . . 152
7.9 Performance of TrustCommNet and Baselines with varying attack intensity. . . . 153
7.10 Scalability of TrustCommNet and Baselines with varying numbers of attackers. . 153
7.11 Results of the ablation studies. . . . 154

8.1 Autonomous fleet management at the strategic decision level. . . . 157
8.2 Four different DRL paradigms. . . . 161
8.3 GridWorld environment. (a) Agent and order representations. (b) Zone-specific parameters. . . . 169
8.4 Results for the GridWorld case study. . . . 170
8.5 Heatmaps of rebalancing actions for the GridWorld case study. . . . 172
8.6 Manhattan environment. (a) Agent and order representations. (b) Passenger pick-up data (1 month). . . . 173
8.7 Total number of passenger pick-ups between 4 and 8 p.m. on five different weekdays. . . . 174
8.8 Comparison of the learning curves for the Manhattan environment (250 agents). 176
8.9 Heatmaps of rebalancing actions for the Manhattan case study. . . . 178
8.10 Average training time per episode for the three learning methods. . . . 179

A.1 Architecture of the EMMA environment. . . . 190
A.2 Classical cooperative environments used for the experiments. . . . 192
A.3 Retraining performance comparison of average team reward. . . . 197
A.4 Response of TrustCommNet to an adversarial attack over time. . . . 197

Nomenclature

Sets

$\mathcal{S}$	Set of states
$\mathcal{A}$	Set of available actions
$\mathcal{O}$	Set of observations
$\mathcal{M}$	Set of messages

Parameters

N	Number of agents
N_{adv}	Number of adversarial agents
γ	Discount factor
θ	Parameters of the neural network
α	Learning rate
ϵ	Adversarial attack intensity
β	Entropy regularization coefficient
λ	Value coefficient
η_{trust}	Trust threshold

Variables

s_i	State of agent i
a_i	Action taken by agent i
o_i	Observation of agent i
r_i	Reward received by agent i
m_i	Message from agent i
m_{adv}	Adversarial message
v	Vehicle speed

Functions

$\mathcal{T}(s'\|s,a)$	Transition probability function
$\mathcal{R}(s,a)$	Reward function
$V(s)$	State value function
$Q(s,a)$	Action-value function
$\mathcal{L}(\theta)$	Loss function
$\nabla_\theta \mathcal{L}$	Gradient of the loss function
$\mathcal{C}(m_i)$	Communication protocol
$\pi_i(s_i)$	Policy of agent i

Chapter 1

Introduction

1.1 Motivation

In recent years, the development of Autonomous Vehicles (AVs) has seen significant advancements, fueled by the promise of enhancing traffic safety and efficiency and reducing human error on the roads. This development is critical, considering that road accidents lead to approximately 1.19 million fatalities and 50 million injuries annually, according to the World Health Organization (World Health Organization, 2023). Technologies such as Advanced Driver Assistance Systems (ADAS), Adaptive Cruise Control (ACC), and Intelligent Transportation Systems (ITS) have been instrumental in mitigating some of these risks (ITS Deployment Evaluation, 2023). However, the ultimate goal remains the deployment of full autonomy, which is expected to drastically reduce the occurrence of traffic accidents by eliminating human errors.

Full AV deployment faces several significant challenges. First, navigating urban environments with unpredictable elements, such as pedestrians and cyclists, remains a formidable task (Koopman and Wagner, 2016). Second, developing robust decision-making algorithms capable of handling diverse and dynamic driving conditions, including adverse weather, is essential. This also involves addressing *edge cases*—rare, atypical scenarios such as unusual road layouts, unpredictable pedestrian movements, erratic human driver behaviour, or sensor malfunctions.

Third, coordinating multiple AVs to avoid collisions and optimise traffic flow in real-time is highly complex (Mariani et al., 2022). Fourth, ensuring cybersecurity to protect against potential hacking and data breaches is critical for safety and maintaining user trust (Bathla et al., 2022). Addressing these and other challenges is vital for realising the full potential of AVs.

Machine Learning has been pivotal in addressing some of the challenges in AV deployment, particularly through Reinforcement Learning (RL). RL has previously demonstrated remarkable achievements in mastering complex games such as Go (Silver et al., 2017a), Chess (Silver et al., 2017b), and various video games (Mnih et al., 2015). These successes have led to extensive applications of RL in autonomous vehicle development (Kiran et al., 2022). However, traditional RL approaches view the problem from a single-agent perspective, which does not account for the need for coordination among multiple AVs or utilise collaborative perception to enhance overall situational awareness. This limitation calls for more advanced frameworks that address system-wide optimisation, enabling better coordination and collaboration among multiple agents in autonomous driving.

To address this limitation and enhance the decision-making capabilities of AVs in more complex environments, Multi-Agent Reinforcement Learning (MARL) emerges as a promising solution. Unlike single-agent RL, MARL involves multiple agents learning to interact within a shared environment, either cooperatively or competitively, to achieve more robust and adaptive policies. This distributed learning framework is particularly suited for scenarios involving numerous AVs and human-driven vehicles, where coordination and real-time decision-making are paramount.

Despite the burgeoning research in MARL for AVs, several challenges remain. First, achieving effective coordination among multiple agents is still a formidable task (Wong et al., 2023). Second, guaranteeing safety in all possible scenarios is not trivial, given the complexity and unpredictability of real-world environments (Muzahid et al., 2023). Third, scaling these algorithms to large instances involving numerous vehicles presents significant computational challenges, such as changing dynamics and learning stability (Papoudakis et al., 2019). Finally, transferring policies learned in simulation to reality, known as sim-to-real transfer, is complex due to the discrepancies between simulated environments and the real world (Candela et al., 2022).

Addressing these challenges is crucial for the practical deployment of MARL in autonomous driving systems.

This thesis addresses the challenges of coordination, safety, scalability, and sim-to-real transfer by organising the research within a hierarchical decision-making framework. These challenges manifest at different levels of autonomous driving—strategic, tactical, and operational—each requiring targeted solutions. The framework serves as an organisational structure, guiding the development of methodologies that target the unique demands of each decision-making level. In the following section, we outline this framework and demonstrate how the proposed methodologies address the corresponding challenges.

1.2 Decision-Making Framework for AVs

Autonomous driving requires vehicles to make decisions across multiple layers of abstraction, from high-level route planning to real-time motion control. These decision-making processes can be divided into three levels (Garrido and Resende, 2022): strategic, tactical, and operational, as presented in Figure 1.1. Each level addresses distinct challenges that are essential for safe and efficient autonomous navigation. MARL is a versatile technique for addressing these challenges by abstracting each task into an agent-based framework. This approach allows for modelling a wide range of problems, from strategic fleet coordination to operational decisions, while supporting decentralised coordination and enhancing adaptability in dynamic environments.

Operational Level: At the operational level, decision-making is focused on real-time, low-level control tasks, such as obstacle avoidance and precise motion control. These decisions are executed on very short timescales and must be made with high precision to ensure safe operation in dynamic environments. Ensuring that policies learned in simulation transfer successfully to real-world scenarios is a key challenge at this level. MARL can improve this process by enabling agents to train and adapt under uncertain conditions, ensuring that real-time planning decisions are robust even in unpredictable real-world environments.

Tactical Level: The tactical level involves medium-term decisions that guide how an au-

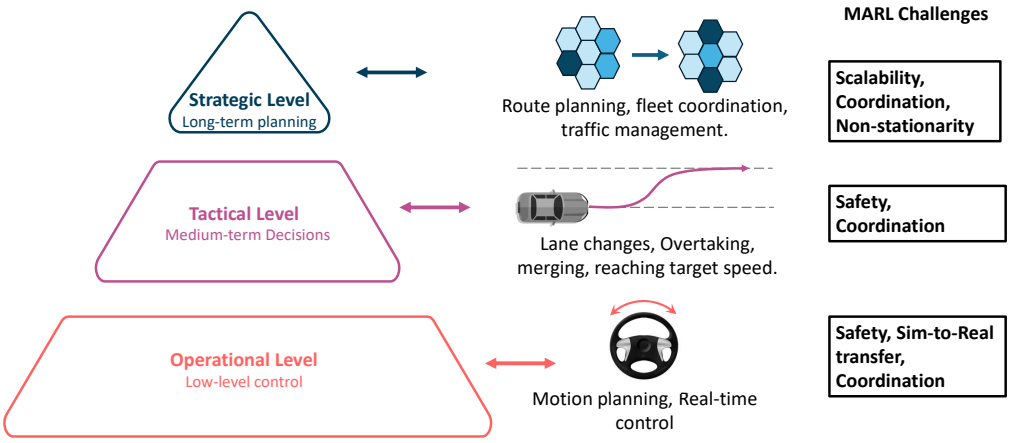


Figure 1.1: Decision-making levels in Autonomous Driving.

tonomous vehicle interacts with its immediate environment. This includes decisions such as lane changes, overtaking, merging and preparing for upcoming traffic conditions. In mixed traffic environments, which often involve human-driven vehicles, the ability of AVs to anticipate and react to others' actions is crucial for ensuring safety and efficiency.

MARL plays a critical role at the tactical level by enabling coordinated decision-making between AVs. Through collaborative communication, AVs can share their intentions and adjust their actions dynamically to ensure safe and efficient navigation. This allows AVs to anticipate the actions of other vehicles and make tactical adjustments that improve overall safety and efficiency.

Strategic Level: At the strategic level, decision-making is concerned with long-term, high-level planning. This includes determining optimal routes, managing traffic, and coordinating fleets of AVs across large-scale environments. The ability to plan routes while considering traffic congestion and other AVs on the road is essential for efficient system-wide mobility.

MARL can enhance strategic decision-making by enabling fleet coordination across multiple AVs, ensuring efficient deployment of vehicles to optimize traffic flow and meet mobility demands. This framework utilises shared information about traffic conditions, vehicle positions, and supply-demand dynamics to optimise collective behaviour over long time horizons, addressing scalability concerns in city-wide autonomous transportation systems.

This decision-making framework establishes the foundation for the rest of this thesis, where MARL solutions are developed to address the challenges at each level. At the operational level, the emphasis is on ensuring robust sim-to-real transfer and effective real-time decision-making. At the tactical level, the research proposes solutions for safe and efficient manoeuvring in complex urban environments. Finally, at the strategic level, the focus is on enhancing fleet coordination and addressing challenges related to scalability. Each of these levels and their associated challenges are addressed in detail through the methodologies and solutions proposed in the following chapters.

1.3 Research Aim and Objectives

The aim of this thesis is to develop and evaluate collaborative, scalable, and safe Multi-Agent Deep Reinforcement Learning solutions to facilitate and enhance the deployment of AVs in real-world environments. The research addresses critical challenges across the various levels of autonomous decision-making, including effective coordination, robust communication, defence against adversarial attacks, scalability, and sim-to-real transfer, ultimately contributing to the creation of reliable, efficient, and secure AV systems.

The specific research objectives are defined as follows:

1. **Coordination and Safety**: Develop MARL methods that ensure coordination and safety across decision-making levels, from low-level control to higher-level manoeuvres, ensuring robustness in both simulated and real-world environments.

2. **Secure and Efficient Communication**: Design robust communication frameworks that enable secure and efficient collaboration among AVs, facilitating navigation in complex urban environments and protecting against adversarial attacks through cooperative defence mechanisms.

3. **Scalability**: Ensure that MARL control mechanisms can scale to large autonomous vehicle systems, addressing the challenges of non-stationarity and coordination in dynamic,

agent-dense environments.

1.4 Thesis Structure

The structure of this thesis is as follows, systematically addressing the different research objectives:

Chapter 2: Presents the mathematical foundations of Single- and Multi-Agent Reinforcement Learning.

Chapter 3: Conducts a comprehensive literature survey on the current methods of MARL for AV systems. It also explores V2X communication, sim-to-real transfer, and traditional methods for optimising traffic, addressing key challenges such as safe navigation and scalability.

Chapter 4: Proposes a method to achieve effective coordination and safety in autonomous vehicles, focusing on sim-to-real transfer to ensure reliable performance in dynamic, real-world environments.

Chapter 5: Develops a method for safe manoeuvring in the presence of emergency vehicles, enhancing coordination among AVs in critical traffic scenarios.

Chapter 6: Develops two efficient communication strategies: one for sharing perception features and another for conveying vehicle intent, aimed at facilitating navigation in complex urban situations.

Chapter 7: Investigates adversarial attacks on AV communication systems and presents an end-to-end learning framework for cooperative defence.

Chapter 8: Examines the scalability of various MARL approaches for autonomous fleet management and proposes a scalable, efficient solution for city-sized applications.

Chapter 9: Provides concluding remarks and discusses potential directions for future research.

The structure of this thesis illustrates the interconnectedness of the chapters in addressing the

multi-level challenges of MARL for autonomous vehicle systems. Chapter 2 lays the mathematical foundations for both Single- and Multi-Agent Reinforcement Learning, which are essential for the methods discussed later. Chapter 3 builds on this by surveying the literature and identifying key challenges, including coordination, scalability, and sim-to-real transfer. Chapter 4 addresses the critical challenge of sim-to-real transfer, proposing a method to ensure that policies learned in simulation can be reliably deployed in real-world environments, achieving effective coordination and safety. Chapter 5 introduces a method for safe manoeuvring of AVs in the presence of emergency vehicles, addressing challenges in complex traffic situations. Chapter 6 enhances these efforts by introducing communication strategies for sharing perception features and vehicle intent, facilitating navigation in intricate urban scenarios.

Chapter 7 examines adversarial attacks on AV communication systems and presents a cooperative defence framework, addressing vulnerabilities that could compromise coordination and communication. Chapter 8 shifts focus to the strategic level, proposing scalable MARL solutions for fleet management in city-sized autonomous mobility systems. Together, these chapters form a cohesive narrative that spans the operational, tactical, and strategic levels, demonstrating how each component advances the field of MARL for autonomous mobility. Table 1.1 provides an overview of the chapters, detailing the specific decision-making levels and the challenges each chapter addresses.

Table 1.1: Thesis methodological chapters, decision-making levels, and challenges.

Chapter	Topic	Decision level	Challenges
4	Sim-to-Real transfer	Operational	Coordination, safety, real-world deployment
5	Manoeuvring around EMVs	Tactical	Coordination, safety, edge-cases
6	Communication strategies	Tactical	Efficient communication, coordination
7	Defence against adversarial attacks	Tactical	Communication integrity, malicious attacks
8	Scalability in fleet management	Strategic	Scalability, coordination, non-stationarity

1.5 Publications related to this work

The research presented in this thesis has led to the following papers:

Papers in international journals:

- Parada, L., Candela, E., Marques, L., & Angeloudis, P. (2023). Safe and efficient manoeuvring for emergency vehicles in autonomous traffic using multi-agent proximal policy optimisation. Transportmetrica A: Transport Science, 0 (0), 1–29.
- Candela, E., Doustaly, O., Parada, L., Feng, F., Demiris, Y., & Angeloudis, P. (2023, July). Risk-aware controller for autonomous vehicles using model-based collision prediction and reinforcement learning. Artificial Intelligence, 320, 103923.
- Ngu, E., Parada, L., Escribano Macias, J. J., & Angeloudis, P. (2022, November). Decentralised Multi-Agent Reinforcement Learning Approach for the Same-Day Delivery Problem. Transportation Research Record: Journal of the Transportation Research Board, 2676 (11), 385–395.

Papers currently under review:

- Parada, L., Yu, K., & Angeloudis, P. IntNet: A Communication-Driven Multi-Agent Reinforcement Learning Framework for Cooperative Autonomous Driving. IEEE Robotics and Automation Letters (RA-L).
- Parada, L., Matsui, K., & Angeloudis, P. Building Trust in Multi-Agent RL: Strategic Defense Against Communication Attacks. AAAI Conference on Artificial Intelligence (AAAI 2025).
- Parada, L., Escribano, J., Feng, Y., Karamanis, R., & Angeloudis, P. Scaling Deep Reinforcement Learning for Autonomous Mobility on Demand Systems: A Curriculum Learning Approach [SSRN Scholarly Paper].

Papers published in conference proceedings:

- Parada, L., Tian, H., Escribano, J., & Angeloudis, P. (2023, September). An End-to-End Collaborative Learning Approach for Connected Autonomous Vehicles in Occluded

Scenarios. In 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) (pp. 5548–5554

- Candela, E., Parada, L., Marques, L., Georgescu, T.-A., Demiris, Y., & Angeloudis, P. (2022, Oc- tober). Transferring Multi-Agent Reinforcement Learning Policies for Autonomous Driving us- ing Sim-to-Real. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 8814–8820).

Academic presentations at international conferences (posters and talks):

- Parada, L. and Angeloudis, P. (2024). Securing autonomous vehicle networks: Strategic defence against communication attacks. In RSS 2024 Workshop for Autonomous Vehicles Across Scale (AVAS), Delft, Netherlands.
- Parada, L., Tian, H., & Angeloudis, P. (2023). Assessing the Robustness of Multi-Agent Deep Reinforcement Learning for Collaborative Navigation under Adversarial V2V Attacks. In 10th International Symposium on Transportation Data & Modelling (ISTDM2023).
- Parada, L., Tian, H., Escribano, J., & Angeloudis, P. (2023). A V2V-based Collaborative Navigation Framework for Connected Autonomous Vehicles in Occluded Scenarios. In the ICRA Workshop on Scalable Autonomous Driving (2023).
- Ngu, E., Parada, L., Escribano Macias, J. J., & Angeloudis, P. (2022, November). Decentralised Multi-Agent Reinforcement Learning Approach for the Same-Day Delivery Problem. Transportation Research Board (TRB) Annual Meeting 2021.

Chapter 2

Mathematical Foundations of Multi-Agent Reinforcement Learning

2.1 Introduction

This chapter explores the mathematical foundations essential for understanding Multi-Agent Reinforcement Learning (MARL). We start by defining Markov Decision Processes (MDPs) for modelling decision-making in uncertain environments, and extend to Partially Observable Markov Decision Processes (POMDPs) for scenarios with incomplete state information.

Next, we explore Stochastic Games (or Markov Games) to generalise MDPs for multi-agent interactions, incorporating game theory to address both cooperative and competitive dynamics. We then introduce Partially Observable Stochastic Games (POSGs) for multi-agent systems with limited state information, which is crucial for applications like autonomous vehicle coordination and collaborative robotics.

Finally, we end with an overview of classical Reinforcement Learning and Multi-Agent Reinforcement Learning methods, including model-free and model-based approaches, value-based methods, policy gradient techniques, and recent advancements in Deep Reinforcement Learning. The modelling frameworks and methods presented in this chapter will serve as building

blocks for developing more complex MARL models for cooperative AV systems in the next chapters.

2.2 Markov Decision Process

A Markov Decision Process (MDP) (Puterman, 1994) provides a mathematical framework for modelling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker or agent. MDPs satisfy the *Markov property*, which means that the future state depends only on the current state and action, not on the sequence of events that preceded it. An MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$:

- **States ($\mathcal{S}$)**: A set of states representing all possible configurations of the system.
- **Actions ($\mathcal{A}$)**: A set of actions available to the agent.
- **Transition Probability Function ($\mathcal{T}$)**: The probability distribution that determines the transition from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ given action $a \in \mathcal{A}$. Formally, $\mathcal{T}(s'|s, a)$.
- **Reward Function ($\mathcal{R}$)**: The immediate reward received after transitioning from state s to state s' due to action a. Denoted as $\mathcal{R}(s, a, s')$.
- **Discount Factor (γ)**: A factor $\gamma \in [0, 1]$ that discounts future rewards, representing the trade-off between immediate and future rewards.

In an MDP, the agent interacts with the environment by taking action and observing the changes in the system's state along with the rewards received. At each time step, the agent is in a state $s \in \mathcal{S}$. The agent selects an action $a \in \mathcal{A}$ based on its policy π, which is a mapping from states to probabilities of selecting each action. Upon executing the action a, the environment transitions to a new state $s' \in \mathcal{S}$ according to the transition probability function $\mathcal{T}(s'|s, a)$. Simultaneously, the agent receives an immediate reward r according to the reward function $\mathcal{R}(s, a, s')$, which quantifies the benefit of the state transition. The process then repeats with

the agent now in the new state s'. The agent selects a new action and continues to interact with the environment. The agent's objective is to maximise the cumulative discounted reward over time, balancing immediate rewards with future benefits, which is governed by the discount factor γ.

2.2.1 Discounted Rewards

The concept of discounted returns is central to MDPs as it encapsulates the agent's objective of maximising long-term rewards while accounting for the time value of rewards. The discounted return G_t at time step t is defined as the sum of future rewards, each multiplied by the discount factor $\gamma \in [0, 1]$ raised to the power of the time step at which the reward is received. Formally, the discounted return is given by:

$$G_t = \sum_{k=0}^{\infty} \gamma^k \mathcal{R}_{t+k+1}, \tag{2.1}$$

where $\mathcal{R}_{t+k+1}$ is the reward received $k+1$ time steps after time t. The discount factor γ ensures that rewards received sooner are valued more highly than those received later, reflecting the principle that immediate rewards are generally preferable. In the context of MDPs, the goal is to find a policy π that maximises the expected discounted return from each state s, which involves determining the optimal action to take in each state to achieve this objective.

2.2.2 Value Functions

The concept of the value function is fundamental to MDPs. A value function $V(s)$ estimates the expected return (total discounted reward) from a given state or state-action pair. It provides a measure of the long-term benefit of being in a particular state and following a certain policy. The purpose of the value function is to quantify the desirability of states, allowing the agent to make informed decisions that maximise future rewards. The value function helps in determining optimal policies by providing a means to compare the long-term effectiveness of different actions.

There are two primary types of value functions: the state value function $V(s)$ and the action-value function $Q(s,a)$. The state value function $V(s)$ represents the expected return starting from state s, while the action-value function $Q(s,a)$ represents the expected return from taking action a in state s. These functions are essential for solving MDPs, as they guide the agent in selecting actions that maximise cumulative rewards over time.

2.2.3 Bellman Equations

The Bellman equations provide a recursive method for computing value functions and are fundamental to many RL algorithms (Bellman, 1957). According to these equations, the value function can be obtained by the following equation:

$$V(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} \mathcal{T}(s'|s,a) \left[\mathcal{R}(s,a,s') + \gamma V(s') \right], \tag{2.2}$$

where $\pi(a|s)$ is the policy function representing the probability of taking action a in state s. Similarly, the Bellman equation for the action-value function $Q(s,a)$ gives the expected return for taking action a in state s:

$$Q(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{T}(s'|s,a) \left[\mathcal{R}(s,a,s') + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s',a') \right]. \tag{2.3}$$

These equations can be extended to define the optimal value functions, which are crucial for determining the best possible strategy an agent can follow. The optimal value function $V^*(s)$ and the optimal action-value function $Q^*(s,a)$ provide the maximum expected return that can be achieved from any state s or state-action pair (s,a), respectively, by following the optimal policy. The Bellman optimality equation for the value function $V^*(s)$ is then:

$$V^*(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{T}(s'|s,a) \left[\mathcal{R}(s,a,s') + \gamma V^*(s') \right]. \tag{2.4}$$

And for the action-value function $Q^*(s, a)$:

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{T}(s'|s,a) \left[\mathcal{R}(s,a,s') + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a') \right]. \tag{2.5}$$

These optimal Bellman equations are critical because they enable the agent to evaluate the long-term benefits of actions and states accurately. By using these optimal functions, the agent can identify the optimal policy π^* that maximises the expected return from any state, ensuring the most effective decision-making process.

2.2.4 Partially Observable Markov Decision Process

Building upon the Markov Decision Process (MDP) framework, a Partially Observable Markov Decision Process (POMDP) generalises MDPs to account for situations where the agent does not have full observability of the states (Kaelbling et al., 1998). This is especially relevant in real-world scenarios where complete state information is often unavailable. A POMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma \rangle$ with two additional elements compared to the MDP definition:

- **Observations ($\mathcal{O}$)**: A finite set of observations that the agent can perceive.
- **Observation Probabilities (Ω)**: The probability of perceiving observation $o \in \mathcal{O}$ given the state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. Formally, $\Omega(o|s', a)$.

2.3 Stochastic Games

A stochastic game (Shapley, 1953), also known as a Markov game, is a generalisation of MDPs to multi-agent scenarios. It is defined by the tuple $\langle N, \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, where:

- N is the number of agents. There are N agents indexed by $i \in \{1, 2, \ldots, N\}$.

- $\mathcal{S}$ is the set of states. The environment can be in one of these states at any given time.
- $\mathcal{A}$ is the set of joint actions: $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_N$, where $\mathcal{A}_i$ is the set of actions available to agent i.
- $\mathcal{T}$ is the state transition probability function: $\mathcal{T}(s' \mid s, a)$ defines the probability of transitioning to state s' from state s when the joint action $a = (a_1, a_2, \ldots, a_N)$ is taken.
- $\mathcal{R}$ is the reward function: $\mathcal{R}_i(s, a, s')$ gives the immediate reward received by agent i when transitioning from state s to state s' as a result of the joint action a.
- γ is the discount factor. $\gamma \in [0, 1]$ represents the discount applied to future rewards, indicating the agents' preference for immediate rewards over future rewards.

In a stochastic game, multiple agents interact within the environment, each aiming to maximise their own cumulative reward. The transition probabilities and rewards depend on the joint actions of all agents, leading to strategic and often competitive interactions.

Stochastic games can also be partially observable (POSGs), where each agent has only limited information about the current state. A special case of POSGs with common rewards, in which all agents share the same reward function, are known as Decentralised Partially Observable Markov Decision Processes (Dec-POMDPs). All the models presented in this thesis are POSGs, with some of them falling in the category of Dec-POMDPs.

2.3.1 Modelling Communication

Stochastic games and POSGs can model communication between agents by incorporating communication actions. These actions are observed by other agents but do not directly affect the environment state. Thus, the action space for agent i can be represented as a combination of environment actions and communication actions (Albrecht et al., 2024):

$$A_i = X_i \times M_i, \tag{2.6}$$

where X_i is a set that contains the environment actions and M_i is a set that contains the communication actions. Messages in the set M_i can be discrete symbols, continuous values, or complex multi-valued vectors. A communication protocol or aggregation function $\mathcal{C} : \mathcal{M}^N \rightarrow \mathcal{M}$ consolidates individual agents' messages into a unified message. This protocol plays a crucial role in exchanging messages, significantly impacting the agents' decision-making process.

2.3.2 Solution Concepts

In the context of stochastic games, particularly POSGs, solution concepts are critical for understanding and predicting agents' behaviour. A solution concept defines the criteria under which agents' strategies are considered optimal or stable. One common solution concept is the *Nash equilibrium*, where no agent can unilaterally improve their expected reward by changing their strategy. Finding a *Nash equilibrium* in POSGs is challenging due to the partial observability and the multi-agent interactions that complicate calculating the best responses. Another important solution concept is the concept of a *best response*, where each agent's strategy is optimal given the strategies of other agents.

In cooperative settings, agents may seek a joint policy that maximises the cumulative reward for all agents, often referred to as the *social welfare* (Albrecht et al., 2024) solution, which is represented as:

$$\pi^* = \arg\max_{\pi} \sum_{i=1}^{n} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_i(s^t, a^t) \mid \pi\right], \tag{2.7}$$

where π denotes the joint policy of all agents, γ is the discount factor, and $\mathcal{R}_i(s^t, a^t)$ is the reward function for agent i at time t. In POSGs and Dec-POMDPs, this involves finding a joint policy that accounts for the limited and noisy observations each agent receives, ensuring that the policy is robust to environmental uncertainties.

This thesis uses the *social welfare* optimal solution concept for all cooperative problems, focusing on finding joint policies that maximise the cumulative rewards for all agents involved.

RL techniques, such as Q-learning and policy gradient methods, are often used to approximate solutions in these complex settings. These methods enable agents to learn optimal or near-optimal policies through repeated interactions with the environment and other agents, gradually improving their decision-making processes even in the presence of partial observability and communication constraints.

2.4 Reinforcement Learning

Reinforcement Learning (RL) is defined as a computational approach to understanding and automating goal-directed learning and decision-making (Sutton and Barto, 2018). Building on the concept of MDPs, RL allows the agent to achieve goals in an uncertain, potentially complex environment by taking actions and observing the outcomes. Unlike supervised learning, RL is based on the idea of trial and error learning (Minsky, 1954; Farley and Clark, 1954), where the agent uses feedback from its own actions and experiences (rewards) to improve its performance over time.

We discuss the differences between model-free and model-based algorithms and highlight popular and effective value-based and policy gradient methods. Subsequently, we introduce extensions of RL, including Deep Reinforcement Learning and Multi-Agent Reinforcement Learning.

2.4.1 Model-free and Model-based RL

RL algorithms can be broadly categorised into model-free and model-based approaches, each offering distinct advantages and trade-offs. Model-free RL algorithms, such as Q-learning and policy gradient methods, learn policies or value functions directly from interactions with the environment without building an explicit model of the environment's dynamics. These methods rely solely on observed rewards and state transitions, which makes them easier to implement but often results in high sample complexity. In contrast, model-based RL algorithms explicitly construct a model of the environment's dynamics, predicting the next state and reward given

the current state and action. This model allows for planning and decision-making, significantly reducing the need for real environment interactions. However, the effectiveness of model-based approaches depends heavily on the accuracy of the learned model; inaccuracies can lead to suboptimal policies. Thus, while model-free methods offer robustness and simplicity, model-based methods provide greater sample efficiency and planning capabilities, often leading to hybrid approaches that combine the strengths of both.

Model-free methods are particularly beneficial for autonomous driving due to their robustness and simplicity, allowing the system to learn effective driving policies directly from interactions without needing an explicit model of the driving environments, which may be extremely complicated. Moreover, in the case of MARL, applying model-based approaches can be even more challenging due to the difficulty in accurately modelling the interactions between multiple autonomous vehicles. In this thesis, all the designed and implemented MARL models fall under the model-free category. The main objective is to learn the value and policy networks directly through interaction with the environment.

2.4.2 Value-based Methods

Value-based methods in RL centre around estimating value functions, which quantify the expected cumulative reward associated with states or state-action pairs. The two primary value-based methods discussed are Q-learning and SARSA.

Q-learning (Watkins and Dayan, 1992) is an off-policy algorithm that updates the value of taking a specific action in a specific state, irrespective of the agent's current policy. The Q-value $Q(s, a)$ denotes the expected return for performing action a in state s. The update rule for Q-learning is derived from the Bellman equation for the optimal action-value function:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[\mathcal{R}(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a) \right], \tag{2.8}$$

where α is the learning rate, R is the reward received after transitioning to state s' from state

s by taking action a, and γ is the discount factor.

SARSA (State-Action-Reward-State-Action) (Sutton and Barto, 2018), on the other hand, is an on-policy algorithm. It updates the Q-value based on the action chosen by the current policy. The update rule for SARSA is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[\mathcal{R}(s,a,s') + \gamma Q(s',a') - Q(s,a)\right], \tag{2.9}$$

where a' is the action taken in the next state s' according to the current policy.

Both Q-learning and SARSA aim to converge to an optimal policy π^* that maximises the expected cumulative reward. They do this by iteratively refining the Q-values based on observed transitions.

2.4.3 Policy Gradient Methods

While value-based methods focus on estimating value functions, policy gradient methods directly optimise the policy itself. By parametrising the policy, these methods use gradient ascent to maximise the expected cumulative reward.

The **REINFORCE** algorithm (Williams, 1992) is a basic policy gradient method that updates the policy parameters θ by following the gradient of the expected return. The update rule for a single sample estimate is:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a|s) G_t, \tag{2.10}$$

where α is the learning rate, $\pi_\theta(a|s)$ is the probability of taking action a in state s under the policy π parameterised by θ, and G_t is the discounted return.

Actor-Critic combine the strengths of value-based and policy gradient methods. They utilise two components: the actor, which updates the policy, and the critic, which evaluates the policy

by estimating the value function. The temporal-difference (TD) error δ^t guides the updates for both the actor and the critic. This error measures the difference between the estimated value of the current state and the observed return, plus the estimated value of the next state:

$$\delta_t = \mathcal{R}_t + \gamma V(s^{t+1}) - V(s^t), \tag{2.11}$$

The update rule for the actor is guided by the gradient of the expected return, scaled by the TD error:

$$\theta \leftarrow \theta + \alpha \delta^t \nabla_\theta \log \pi_\theta(a^t|s^t), \tag{2.12}$$

The critic updates its parameters w to reduce the TD error by adjusting the estimated value function:

$$w \leftarrow w + \beta \delta^t \nabla_w V(s^t), \tag{2.13}$$

β is the learning rate for the critic. Policy gradient methods are advantageous in high-dimensional or continuous action spaces because they optimise the policy directly rather than approximating a value function for each action. These methods are applied in various complex RL problems, including robotics, game playing, and autonomous driving, where direct policy optimisation is crucial for performance.

2.4.4 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) combines the principles of RL with Deep Learning to handle high-dimensional state and action spaces. Traditional RL algorithms often struggle with such complexity due to the curse of dimensionality. The curse of dimensionality refers to the exponential increase in computational and data requirements with the number of state

or action variables. DRL addresses this challenge by utilising neural networks as function approximators to represent value functions, policies or both.

One of the pioneering algorithms in DRL is the Deep Q-Network (DQN) (Mnih et al., 2016). DQN leverages a deep neural network to approximate the Q-value function, $Q(s, a; \theta)$, where θ represents the parameters of the neural network. The network is trained using a variant of the Q-learning update rule:

$$\theta \leftarrow \theta + \alpha \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right] \nabla_\theta Q(s, a; \theta), \tag{2.14}$$

where θ^- are the parameters of a target network, which are periodically updated to stabilise training. The use of experience replay, where transitions (s, a, R, s') are stored in a replay buffer and sampled randomly during training, helps to break the correlation between consecutive updates, improving the stability and efficiency of the learning process.

Another significant advancement in DRL is the introduction of actor-critic methods with deep learning, such as Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015). DDPG is designed for continuous action spaces and combines the benefits of policy gradient methods with function approximation. It consists of two networks: the actor, which determines the policy $\pi(s; \theta^\mu)$, and the critic, which estimates the Q-value $Q(s, a; \theta^Q)$. The critic is trained using the temporal-difference error:

$$\delta_t = \mathcal{R}(s, a, s') + \gamma Q(s', \pi(s'; \theta^\mu); \theta^Q) - Q(s, a; \theta^Q), \tag{2.15}$$

and the actor is updated by following the policy gradient:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s \sim \mathcal{D}} \left[\nabla_a Q(s, a; \theta^Q) \big|_{a = \pi(s; \theta^\mu)} \nabla_{\theta^\mu} \pi(s; \theta^\mu) \right], \tag{2.16}$$

where $\mathcal{D}$ is the replay buffer.

DRL has demonstrated remarkable success in various domains, including playing complex games (Silver et al., 2017a), video games (Mnih et al., 2015), and autonomous driving (Kiran et al., 2022). By leveraging the representation power of deep neural networks, DRL algorithms can learn directly from raw sensory inputs, enabling the agent to perform well in environments with high-dimensional observations and actions.

2.4.5 Multi-Agent Reinforcement Learning

Multi-agent Reinforcement Learning (MARL) extends the principles of RL to environments where multiple agents interact. These agents can cooperate, compete, or both, leading to complex dynamics not addressed by single-agent RL. MARL algorithms aim to find optimal policies for each agent while considering the presence and actions of other agents. MARL is closely related to stochastic games and POSGs, including Dec-POMDPs, where agents must learn and adapt in settings with both cooperation and competition under uncertainty.

One popular MARL algorithm is Multi-Agent Deep Deterministic Policy Gradient (MADDPG) (Lowe et al., 2020), which is an extension of the DDPG algorithm for multi-agent environments. MADDPG uses a centralised training and decentralised execution (CTDE) framework. This means that each agent has its own actor network, while a centralised critic network is used during training to estimate the Q-value, incorporating the actions of all agents to capture their interactions.

For each agent i, the actor network π_i determines the policy $\pi_i(s_i; \theta_i^\mu)$, and the centralised critic Q_i estimates the Q-value $Q_i(x, a_1, \ldots, a_N; \theta_i^Q)$, where x represents the state information and a_j denotes the actions of all agents j. The critic is updated using the temporal-difference error for the multi-agent case:

$$\delta_i = \mathcal{R}_i + \gamma Q_i(x', a'_1, \ldots, a'_N; \theta_i^Q) - Q_i(x, a_1, \ldots, a_N; \theta_i^Q), \tag{2.17}$$

where $a'_j = \pi_j(s'_j; \theta_j^\mu)$ for all agents j. The actor for agent i is updated by following the policy

gradient:

$$\nabla_{\theta_i^\mu} J_i \approx \mathbb{E}_{x,s_i\sim\mathcal{D}} \left[\nabla_{a_i} Q_i(x, a_1, \ldots, a_N; \theta_i^Q)\big|_{a_i=\pi_i(s_i;\theta_i^\mu)} \nabla_{\theta_i^\mu} \pi_i(s_i; \theta_i^\mu) \right], \tag{2.18}$$

This equation indicates that the policy parameters θ_i^π are adjusted in the direction that increases the expected Q-value. Here, $\nabla_{a_i} Q_i(x, a_1, \ldots, a_N; \theta_i^Q)\big|_{a_i=\pi_i(s_i;\theta_i^\mu)}$ represents the gradient of the Q-value with respect to the action a_i, evaluated at the current policy's action. The term $\nabla_{\theta_i^\pi} \pi_i(s_i; \theta_i^\pi)$ is the gradient of the policy function with respect to its parameters. The expectation $\mathbb{E}_{x,s_i\sim\mathcal{D}}$ is taken over the state-action pairs sampled from the replay buffer $\mathcal{D}$, which contains experiences from all agents. This process ensures that the actor updates are based on a wide range of past experiences, improving learning stability and efficiency.

Another popular MARL algorithm is QMIX (Rashid et al., 2018b), a value-based algorithm designed for cooperative multi-agent tasks. It leverages a combination of individual Q-value functions and a mixing network to estimate a joint Q-value. QMIX ensures that the joint Q-value is a monotonic function of the individual Q-values, facilitating efficient learning in cooperative settings.

In QMIX, each agent i has its own Q-network $Q_i(s_i, a_i; \theta_i)$. These individual Q-values are combined using a mixing network that produces the joint Q-value $Q_{tot}(s, \mathbf{a}; \theta)$, where s is the global state and $\mathbf{a}$ is the joint action vector. The mixing network is parameterised by θ and ensures monotonicity by constraining the weights to be non-negative. The joint Q-value is used to compute the temporal-difference error for updating the individual Q-networks:

$$\delta = \mathcal{R}(s, \mathbf{a}, s') + \gamma \max_{\mathbf{a}'} Q_{tot}(s', \mathbf{a}'; \theta) - Q_{tot}(s, \mathbf{a}; \theta), \tag{2.19}$$

where $Q_{tot}(s, \mathbf{a}; \theta) = f(Q_1(s_1, a_1; \theta_1), \ldots, Q_N(s_N, a_N; \theta_N); \theta)$ and f denotes the mixing network function. The individual Q-networks are updated to minimise the temporal-difference error, ensuring that the learning process is aligned with the cooperative objective of maximising the joint Q-value.

2.5 Concluding Remarks

This chapter presented the mathematical foundations behind Reinforcement Learning and Multi-Agent Reinforcement Learning. We explored key concepts, including Markov Decision Processes, Partially Observable Markov Decision Processes, and Stochastic Games, which provide the theoretical underpinnings for decision-making in single-agent and multi-agent settings. Additionally, we discussed classical RL and MARL methods, highlighting model-free approaches, value-based methods, policy gradient methods, and advancements in Deep Reinforcement Learning. The following chapters will build upon these concepts to develop models that address the challenges of MARL in the context of autonomous driving, aiming to enhance the coordination and decision-making capabilities of autonomous vehicles.

Chapter 3

Literature Review

3.1 Introduction

The advancement of cooperative autonomous driving is fundamental for the future of transportation, promising enhanced safety, efficiency, and convenience. This literature review encompasses key research areas essential to developing such systems, including traffic optimisation, risk-aware systems, intersection management, intelligent signal control, and Multi-Agent Reinforcement Learning (MARL). Additionally, it addresses the critical aspect of sim-to-real transfer to bridge the gap between simulated environments and real-world applications.

The chapter begins by examining fundamental aspects of Autonomous Vehicle (AV) systems, including traffic flow optimisation and risk-aware decision-making. It explores various approaches to autonomous intersection management, intelligent signal control, and multi-agent collaboration, highlighting both heuristic and machine learning methods. Next, it addresses coordination among AVs, particularly in the presence of emergency vehicles, by reviewing methodologies for risk-aware autonomous systems and intelligent signal control that ensure safe manoeuvring.

The chapter then delves into communication strategies within MARL frameworks, reviewing recent advancements in collaborative perception and multi-agent communication, which are crucial for effective navigation in complex urban environments. Additionally, it examines ad-

versarial attacks on AV communication systems, highlighting vulnerabilities and state-of-the-art defence mechanisms.

Scalability poses a significant challenge in deploying MARL for autonomous driving. The literature on this topic is analysed, focusing on scalable MARL approaches for autonomous ridesharing and large-scale applications. Various fleet management strategies, including traditional methods and DRL-based solutions, are evaluated for their effectiveness and efficiency.

Finally, the chapter addresses the *sim-to-real gap*, a critical barrier to real-world implementation of MARL policies. It discusses various domain adaptation techniques and solutions for transferring learned policies from simulation to real-world environments.

This comprehensive literature survey sets the stage for the subsequent chapters, which propose novel methods and strategies to address these challenges and advance MARL for AV systems.

3.2 Autonomous Vehicle Systems

In this section, we present the relevant literature on autonomous traffic systems, classified into the following subtopics: Optimising Autonomous Traffic, Risk-aware Autonomous Systems, Autonomous Intersection Management, Risk Metrics, Simulation Tools, MARL for Autonomous Driving, and Gym Environments.

3.2.1 Optimising Autonomous Traffic

Optimising autonomous traffic is crucial for enhancing overall traffic flow, reducing congestion, and improving safety on the roads. Several studies have been conducted on optimising autonomous driving in traffic, particularly in highway scenarios (Claussmann et al., 2020). These studies have focused mostly on single-agent navigation, where traffic is considered part of the environment. Li et al. (2016) proposed a game-theoretic traffic model and evaluated two single-agent policies based on Stackelberg games and Decision Trees. Deep Reinforcement Learning

(DRL) has also been applied to generate high-level policies for the ego vehicle (Nageshrao et al., 2019). A similar study by Ngai and Yung (2011) presented a Reinforcement Learning-based method that considers seven different autonomous driving goals in order to guarantee safe and efficient overtaking manoeuvres. The Q-learning algorithm has also been used to develop efficient single-agent lane-changing policies (Wang et al., 2018). Wang et al. (2021b) developed a multi-agent game-theoretic planning approach for competitive racing scenarios. In this approach, while the agent must collaborate to avoid collisions, its overall goal remains to outperform competitors. The study employs an algorithm to solve for the approximate Nash Equilibrium, assuming that each vehicle has complete knowledge of the intentions and objective functions of all other vehicles. While single-agent methods have made significant strides in optimising autonomous traffic, the growing complexity of multi-agent interactions highlights the need for more advanced, cooperative strategies, which are crucial for real-world deployment.

3.2.2 Risk-aware Autonomous Systems

Designing risk-aware autonomous systems is essential for ensuring the safety and reliability of autonomous vehicles in unpredictable environments with multiple agents. A recent study (Zhang et al., 2021b) used the soft actor-critic (SAC) algorithm and control theory-based Lyapunov functions to train an agent subject to safety constraints. Wen et al. (2020) proposed Parallel Constrained Policy Optimisation (PCPO), which consists of three neural networks to estimate the policy function, value function, and a risk function. They use constraints and lack explicit risk models that capture risk when constraints are not violated. In another study by Nyberg et al. (2021), risk metrics were coupled with a sampling-based trajectory planner. Trajectory planners require a probabilistic model to sample from, which is complex to obtain and can be overcome with model-free RL approaches. An RL method by Li et al. (2016) introduced hard constraints for Autonomous Driving, where cars are not allowed to take certain actions, e.g., change lanes when there's a car on the other lane. Similarly, Nageshrao et al. (2019) introduced an explicit short-horizon safety check to avoid making catastrophic decisions.

In the literature on risk-aware autonomous systems, the subfield of RL that focuses specifically

on safety as a critical objective is known as Safe RL. Garcıa and Fernández (2015) segmented the early literature into two categories: modification of the optimisation criteria and modification of the exploration process. In some approaches, risk functions are learned during training (Geibel and Wysotzki, 2005; Garcia and Fernández, 2012), while in most cases, they are applied during testing. Recent works have focused on Safe RL for single-agent navigation, including studies on lane-changing manoeuvres (Krasowski et al., 2020) and highway scenarios (Kalweit et al., 2020; Wang, 2022; Cao et al., 2023). To the best of our knowledge, no work has been conducted on Safe Multi-agent RL for autonomous vehicles that incorporate explicit risk functions as part of the optimisation criteria.

3.2.3 Autonomous Intersection Management

Efficient intersection management is a crucial aspect of AV systems, as it directly impacts traffic flow, safety, and overall transportation efficiency. Intersections are critical points in traffic networks where the paths of multiple vehicles converge, often leading to congestion and accidents. Autonomous intersection management aims to optimise the movement of vehicles through these junctions, reducing delays, minimising collisions, and enhancing the overall efficiency of transportation networks.

Heuristics and Rule-based methods

The majority of studies in the field of intersection management rely on heuristics and rule-based methods. Alonso et al. (2011) introduced two rule-based priority methods for mixed scenarios, including those with manually driven cars. Following this, Zohdy et al. (2012) proposed a cooperative autonomous cruise control system designed to optimise each vehicle's arrival time at the intersection stop-line, reducing the average delay by 91% compared to traditional intersection control methods. Similarly, Hafner et al. (2013) developed an efficient communication-based algorithm that generates a control throttle/brake map to prevent collisions between vehicles.

Other popular rule-based methods include reservation systems (Au et al., 2015; Vasirani and

Ossowski, 2012; Zhang et al., 2013; Wei et al., 2014) and Time-To-Collision (TTC) frameworks (Minderhoud and Bovy, 2001; van der horst and Hogema, 1994). Reservation systems have been widely used and involve an intersection manager that simulates vehicle trajectories and assigns reservation slots to vehicles approaching the intersection, achieving superior performance in many scenarios. TTC, used extensively as a safety indicator, has also demonstrated superior performance compared to other rule-based methods in ensuring vehicle safety at intersections (Minderhoud and Bovy, 2001).

Studies focusing on unsignalised intersections have also used various heuristics and rule-based methods. For instance, Lu and Kim (2017) applied Genetic Algorithms, while Zhu and Ukkusuri (2015) utilised Linear Programming to determine the optimal passing sequence of AVs. Miculescu and Karaman (2016) employed Polling Systems, and Zhang et al. (2016) proposed a Decentralised Optimal Control method for this purpose.

Rule-based and heuristic methods can be simple and easy to implement; thus, they are suitable for real-time intersection control. However, the complexity of these methods can substantially increase when adding more vehicles and constraints (Namazi et al., 2019). A busy intersection may have several CAVs, human-driven cars and pedestrians. All of these entities can arrive at different times and velocities; consequently, the number of potential scenarios is enormous. TTC-based and other rule-based methods cannot possibly cover all scenarios that can occur in a busy intersection. Furthermore, occluded intersection scenarios can be particularly challenging for traditional methods that do not rely on any form of collaboration.

Deep RL-based methods

The use of DRL has shown promise in enhancing autonomous driving performance in challenging intersection scenarios where traditional rule-based methods may be insufficient. For instance, Kamran et al. (2020) proposed a DQN approach that penalises risky situations resulting from occlusion. Their study demonstrated that the DQN risk-aware method could effectively learn in complex scenarios, while collision-based DQN and rule-based methods failed. Similarly, Isele et al. (2018) also used the DQN algorithm but relied on an occupancy grid to model

Table 3.1: Summary of Literature on Autonomous Intersection Management Methods

Method	Study	Approach	Key Features
Heuristic	Alonso et al. (2011)	RPM	Priority methods for mixed scenarios
Rule-based	Zohdy et al. (2012)	CACC	Reduces average delay by 91%
Rule-based	Hafner et al. (2013)	CBA	Generates control map to prevent collisions
Rule-based	Au et al. (2015)	RS	Intersection manager assigns reservation slots
Rule-based	Vasirani and Ossowski (2012)	RS	Combinatorial auctions
Rule-based	Zhang et al. (2013)	RS	Centralised scheduling algorithm, high-priority vehicles
Rule-based	Wei et al. (2014)	RS	Greedy conflict matrix reservation
Rule-based	Minderhoud and Bovy (2001)	TTC	Develops two TTC-based safety indicators
Rule-based	van der horst and Hogema (1994)	TTC	Similar to above
Heuristic	Lu and Kim (2017)	GA	Optimal passing sequence, Emergency Vehicles
Heuristic	Zhu and Ukkusuri (2015)	LP	Guarantees safety
Heuristic	Miculescu and Karaman (2016)	Polling System	Guarantees safety, stochastic arrival times
Heuristic	Zhang et al. (2016)	DOC	Determines optimal passing sequence
DRL	Kamran et al. (2020)	DQN	Penalises risky situations resulting from occlusion
DRL	Isele et al. (2018)	Occupancy Grid	Models occluded area from the ego vehicle's perspective
DRL	Tram et al. (2018)	MPL	Predicts vehicle intent, negotiates policies

RPM - Rule-based Priority Methods, CACC - Cooperative Autonomous Cruise Control, CBA - Communication-based Algorithm, RS - Reservation System, TTC - Time-To-Collision, GA - Genetic Algorithms, LP - Linear Programming, DOC - Decentralised Optimal Control, DQN - Deep Q-Network, MPL - Metadata-based Policy Learning.

the occluded area from the ego vehicle's perspective. They showed that their method could navigate successfully in scenarios where rule-based methods consistently failed. Additionally, Tram et al. (2018) used metadata of nearby vehicles to predict vehicle intent and learn a policy that can negotiate, thereby demonstrating the potential of machine learning methods for improving autonomous driving performance in challenging scenarios.

Most of the DRL studies have approached the occluded navigation problem from a single-agent perspective (Isele et al., 2018; Kamran et al., 2020; Tram et al., 2018). However, challenging occluded scenarios require active collaboration from CAVs in order to guarantee safe and efficient navigation. The integration of collaborative perception and decision-making mechanisms can enhance the capability of CAVs to share knowledge about the environment and take coordinated actions. A summary of the literature on Autonomous Intersection Management is presented in Table 3.1.

Intersection Management in Presence of Emergency Vehicles

Managing intersections efficiently becomes even more critical in the presence of Emergency Vehicles (EMVs), where the need for swift and safe passage is paramount. Addressing this

Table 3.2: Summary of literature on the passing of EMVs in traffic.

Study	Approach	Autonomous Traffic	Rule-based	Scenario
Lu and Kim (2017)	Genetic Algorithm	✓	x	Junction
Humayun et al. (2022)	Priority rules	✓	✓	Junction
Nellore and Hancke (2016)	Distance-based algorithm	x	✓	Junction
Buckman et al. (2021)	Game-theoretic	✓	x	Highway
Karmakar et al. (2021)	Priority rules	x	✓	Junction
Shelke et al. (2019)	Priority ITS	x	✓	Junction
Dresner and Stone (2006)	Reservation-based	✓	✓	Junction
González et al. (2021)	Priority rules	✓	✓	Junction

challenge is essential for reducing response times in critical emergencies, making it a key focus for intelligent urban transportation systems. Extensive work has been done regarding Intelligent Traffic Signal (ITS) control in the presence of EMVs, focusing on determining the best green light sequence. Chowdhury (2016) present an ITS method to optimise the travel time of EMVs that considers a priority system based on the type of incident. Similarly, Karmakar et al. (2021) uses a priority-based system to estimate the number of signal interventions while considering the impact on traffic in the surrounding areas. Virtual Traffic Lights (VTL) have also been used for efficient crossing of EMVs in autonomous traffic intersections (Viriyasitavat and Tonguz, 2012). In VTL, an AV acts as a leader and manages the traffic by serving as a virtual traffic light for other vehicles. Another study by Shelke et al. (2019) combines a message-passing approach with a priority-based traffic signal method that minimises the delay of data packages. A method for scheduling both green light sequence and duration is proposed by Nellore and Hancke (2016). Their method is based on sensing information, such as the distance to the intersection and the number of cars on the road. Two articles (So et al., 2020; Su et al., 2021) have provided joint methods to optimise traffic signals and vehicle movements. Su et al. (2021) couple a Decentralised RL method to control the traffic signals with a dynamic Dijkstra's algorithm to find the optimal routes.

3.2.4 Risk Metrics

In the domain of autonomous driving, risk metrics are pivotal for evaluating the safety and performance of AVs. These metrics quantify the likelihood and severity of critical events,

allowing engineers to develop and test safer AV systems. As AV technology advances, the complexity of driving environments and the unpredictability of other road users necessitate the development of increasingly sophisticated risk metrics. By combining traditional safety assessment approaches with probabilistic and AI-driven models, engineers can create a more holistic framework for evaluating and mitigating risks in real time.

This subsection explores various categories of risk metrics. Time-based metrics provide foundational approaches to measuring collision risk, focusing on factors like time-to-collision and evasive manoeuvres. Threat-centric and probabilistic metrics extend the assessment by evaluating specific driving actions and applying probabilistic models to anticipate accident scenarios. Finally, AI-based methodologies leverage machine learning techniques to adapt to the dynamic and uncertain nature of real-world environments, offering advanced capabilities in risk prediction and management.

Time-based Metrics

One of the most widely used metrics is Time-to-Collision (TTC) (Hayward, 1972), which measures the time remaining before a potential collision occurs if both vehicles maintain their current speed and trajectory. Although TTC is a useful metric in simple scenarios, its limitations become apparent in more complex environments, as it fails to account for evasive manoeuvres. To address this, researchers have developed variations such as Potential Time-to-Collision (PTTC), which incorporates potential future manoeuvres and adjusts the collision prediction accordingly (Wakabayashi et al., 2003).

Several other time-based metrics enhance risk assessment by considering different aspects of vehicle interactions. For instance, Time to Closest Encounter (TTCE) measures the minimum time before vehicles reach their closest point of approach, providing an indication of imminent danger even when a collision is not likely (Eggert, 2014). Time to Zebra (TTZ), which focuses on the time until an AV reaches a pedestrian crossing, is another specialised metric designed for urban environments (Cafiso et al., 2011). Additionally, Time to Manoeuvre (TTM) estimates the time required for an AV to complete an evasive manoeuvre, considering its current speed

and direction (Hillenbrand et al., 2006).

Threat-Centric and Probabilistic Metrics

Other risk metrics include Brake Threat Number (BTN), which evaluates the likelihood of a situation requiring emergency braking (Jansson, 2005), and Steer Threat Number (STN), which focuses on the potential need for sudden steering corrections to avoid hazards (Eidehall, 2011). Both metrics are part of a broader category of "threat numbers" that assess specific vehicle actions needed to avert accidents. Furthermore, metrics like Crash Potential Index (CPI) (Cunto and Saccomanno, 2008) and Conflict Severity (CS) (Bagdadi, 2013) offer a probabilistic approach, estimating the likelihood of a crash based on current traffic conditions and vehicle trajectories.

Finally, hybrid models integrating both traditional risk assessment techniques and probabilistic simulations are becoming crucial tools in AV safety. For example, Collision Probability via Monte Carlo (P-MC) (Broadhurst et al., 2005) and Stochastic Reachable Sets (P-SRS) (Althoff et al., 2009) utilise simulation-based approaches to predict possible collision outcomes, adding another layer of probabilistic analysis. These models help account for uncertainties in both vehicle behaviour and environmental factors, making them highly adaptable to real-world conditions.

AI-based methodologies

AI-based methodologies are increasingly important in risk assessment for autonomous driving, particularly for their ability to handle dynamic and unpredictable environments. These methodologies leverage Deep Learning to model complex interactions between vehicles and their surroundings, enabling more accurate risk predictions. For instance, one notable approach uses convolutional neural networks (CNN) to process real-time visual data and assess potential risks, such as collisions, by dividing the scene into sections of varying risk levels (e.g., normal, caution, warning) and using risk metrics like grid-based occupancy calculations (Feth et al.,

2018). Another approach employs Bayesian Convolutional Long Short-Term Memory (ConvLSTM) networks to incorporate temporal information, improving predictive capabilities in environments with high uncertainty. This method evaluates the probability of future collisions, integrating information from multiple modalities such as radar and camera inputs (Strickland et al., 2018).

A more recent deep learning-based model applies *looming*, a method that estimates whether surrounding objects are on a collision course with the ego-vehicle. By incorporating multiple factors beyond TTC, such as vehicle kinematics and macroscopic traffic data, this approach increases the sensitivity of threat detection under uncertainty, addressing the limitations of traditional models that rely on single factors (Formosa et al., 2022). Despite their advantages, AI-driven models still face significant challenges related to uncertainty (Chia et al., 2022), as deep learning systems often exhibit both aleatoric uncertainty (due to inherent randomness in the environment) and epistemic uncertainty (stemming from a lack of knowledge or data).

3.2.5 Simulation Tools

Simulation tools are essential for the development and testing of AV systems, providing a safe, scalable, and cost-effective environment for evaluating algorithms before real-world deployment. High-fidelity simulators like CARLA (Dosovitskiy et al., 2017) and the SVL Simulator (Rong et al., 2020) offer detailed 3D environments, realistic physics, and sensor integration, making them ideal for testing perception, planning, and control in various driving conditions. CARLA is a notable 3D autonomous driving simulator known for its realistic physics model and high flexibility. It allows users to easily modify onboard sensors, such as cameras or LiDAR, and simulate different weather conditions, making it a valuable tool for developing and evaluating autonomous driving systems.

Traffic simulation tools are vital for evaluating AV behavior in complex traffic environments. SUMMIT (Cai et al., 2020) integrates large-scale traffic simulations with vehicle dynamics to test AV systems in mixed-traffic conditions. Similarly, simulators like SUMO (Lopez et al.,

2018) and VISSIM (PTV Group, 1993) play an essential role in traffic optimisation and intersection management. These tools allow researchers to study interactions between human-driven and autonomous vehicles, facilitating the integration of AVs into existing urban traffic systems.

For vehicle dynamics, CarSim (Mechanical Simulation Corporation, 2023) and IPG CarMaker (IPG Automotive, 2023) stand out as advanced simulators offering high-fidelity modelling of vehicle behaviour under various driving conditions. These tools are particularly valuable for testing control algorithms and optimising vehicle dynamics due to their detailed models of suspension, powertrain, and tire-road interactions. Webots (Michel, 2004) is another simulator known for its multi-robot simulation capabilities and integration with physics engines, making it useful for testing vehicle dynamics in robotic and autonomous vehicle applications.

In sensory data simulation, AirSim (Shah et al., 2018) and DeepDrive (DeepDrive Team, 2019) offer highly realistic sensor models and environmental dynamics, simulating a variety of cameras, LiDAR, and radar in diverse weather and lighting conditions. These simulators are indispensable for developing robust perception algorithms, particularly when real-world data is scarce or costly to obtain. Gazebo (Koenig and Howard, 2004), a robotics simulator, also supports realistic physics-based simulations of sensor data and vehicle dynamics, making it a versatile tool for both perception and control tasks.

Hybrid frameworks like Apollo (Baidu, 2017) and Autoware (Autoware Foundation, 2018) provide open-source platforms for developing and testing autonomous vehicle software, focusing on integrating perception, planning, localisation, and control. While they do not directly simulate vehicle dynamics or traffic at scale, they are often paired with simulators like CARLA or SVL to enable comprehensive testing of AV systems. These frameworks offer modular components that facilitate the end-to-end development of autonomous driving software and help bridge the gap between simulation and real-world applications by allowing seamless integration with both hardware-in-the-loop and software-in-the-loop testing environments.

3.2.6 MARL for Autonomous Driving

MARL has been extensively explored in autonomous driving, focusing on collaborative navigation, path planning, traffic signal control, and addressing safety and coordination challenges in dynamic urban settings. Figure 3.1 illustrates the MARL approach for autonomous driving, where agents interact with the environment through actions, observations, and rewards, guided by the policy. A few survey works have helped categorise the literature (Dinneweth et al., 2022; Yadav et al., 2023). Recent efforts have demonstrated MARL's efficacy in facilitating collaborative navigation and path planning (Candela et al., 2022; Shalev-Shwartz et al., 2016; Troullinos et al., 2021; Adan et al., 2023) and traffic signal control (Wang et al., 2021c; Hu et al., 2023; Wang et al., 2023). These developments underline MARL's capacity to improve traffic efficiency through intelligent agent collaboration.

Further advancements in MARL for autonomous traffic management include addressing complex scenarios such as the double-merging problem (Shalev-Shwartz et al., 2016), cooperative lane-changing in the presence of human-driven Vehicles (Zhou et al., 2022), and highway ramp-merging strategies (Chen et al., 2022). Key research areas focus on using Graph Convolutional Neural Networks (GCN) to enhance manoeuvring and safety (Dong et al., 2020) and applying MARL in challenging settings like near Emergency Vehicles (Parada et al., 2023a) and in occluded situations (Parada et al., 2023d). Despite these promising efforts, safety and coordination in dynamic urban environments persist as largely unresolved issues in MARL for autonomous driving, underscoring the need for innovative solutions to address the unpredictability and complexity of these scenarios (Dinneweth et al., 2022).

3.2.7 Gym Environments

Designing and modelling environments for MARL is crucial for evaluating and improving learning algorithms under realistic conditions. To facilitate this, several studies have introduced autonomous driving gym environments, which are simulation frameworks that allow agents to interact, receive feedback, and improve their policies in a controlled, replicable setting.

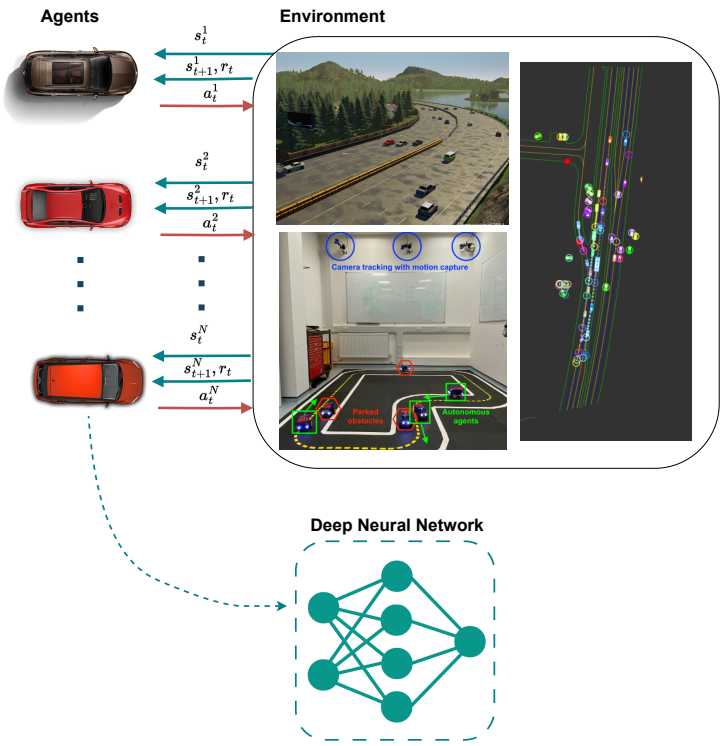


Figure 3.1: Schematic overview of the MARL approach to autonomous driving: Agents interact with the environment by taking actions and receiving observations and rewards. The environment could be a simple 2D simulator, a high-fidelity 3D simulator or a robotic testbed. Actions are guided by a policy, which can be represented as a neural network.

Some environments use simplified dynamics to enhance parallelisation and evaluation efficiency. For instance, SMARTS (Zhou et al., 2020a) is a 2D multi-agent gym environment for AVs, covering various traffic situations such as lane merging, intersections, overtaking, and roundabouts. It incorporates different controllers, baselines, and several evaluation metrics. Similarly, Highway-env (Leurent, 2018) contains a collection of single-agent lightweight environments for autonomous driving, easily extendable to multi-agent settings, and has been widely used in AV research over the past five years due to their simple and intuitive architecture. Scenario Gym (Scott et al., 2023) offers a high level of customisation, allowing users to tailor various components such as vehicle intelligence and physical models. Nocturne (Vinitsky et al., 2022) is also a 2D simulator that allows to use of expert data (currently designed for the Waymo Open Dataset) and simulates state-based partial observability.

Other gym environments have been created on top of high-fidelity simulators such as CARLA (Dosovitskiy et al., 2017). For instance, Palanisamy (2019a) presented the Multi-Agent Con-

nected Autonomous Driving Gym environment (MACAD), built on top of CARLA, which supports training various Deep RL algorithms in intersection scenarios. MACAD features diverse agent types (homogeneous/non-homogeneous), task natures (cooperative/competitive), and observation modes (full/partial). Another simulator, SUMMIT (Cai et al., 2020), also built on top of CARLA, generates large instances of mixed traffic with realistic, unregulated driving behaviours, though it lacks OpenAI Gym integration. MetaDrive (Li et al., 2022), built on top of the Panda3D graphics engine (Goslin and Mine, 2004), is a highly efficient multi-agent 3D gym environment capable of generating infinite scenarios, making it a versatile tool for testing and training autonomous driving systems.



(a) Highway-env (Leurent, 2018) (b) MACAD (Palanisamy, 2019a) (c) SMARTS (Zhou et al., 2020a)

Figure 3.2: Visual representation of three popular autonomous driving gym environments.

Another popular multi-agent environment is the Gym-Duckietown (Chevalier et al., 2018), which replicates the Duckietown autonomous driving testbed and offers a lane-following environment for testing various RL methods. Figure 3.2, the Graphical Unit Interface (GUI) of three of these gym environments. In addition, a summary of the main characteristics of the different gym environments is presented in Table 3.3.

3.3 Multi-agent Communication and Adversarial Attacks

This section delves into the critical aspects of multi-agent systems and the emerging threats posed by adversarial attacks. It explores the foundational strategies in MARL with commu-

Table 3.3: Summary of the main gym environments for autonomous driving simulation.

Environment	Multi-agent Support	2D/3D	OpenAI Gym Integration	Expert Data
SMARTS (Zhou et al., 2020a)	✓	2D	✓	
Highway-env (Leurent, 2018)		2D	✓	
Scenario Gym (Scott et al., 2023)		2D		✓
Nocturne (Vinitsky et al., 2022)	✓	2D	✓	✓
CARLA (Dosovitskiy et al., 2017)		3D		
MACAD (Palanisamy, 2019a)	✓	3D	✓	✓
SUMMIT (Cai et al., 2020)		3D		
MetaDrive (Li et al., 2022)	✓	3D	✓	
Duckietown (Chevalier et al., 2018)	✓	2D	✓	

nication, the Machine Theory of Mind (MToM) concept for enhanced cooperation, the application of collaborative perception in autonomous vehicles, and the vulnerabilities introduced by adversarial attacks on general deep learning systems and Vehicle-to-Everything (V2X) communications. A summary of the V2X technologies and the current standards can be found in Appendix C.

3.3.1 MARL with Communication

Recent advancements in MARL emphasise the critical role of efficient communication strategies among agents to achieve cooperative objectives while minimising bandwidth usage. A comprehensive survey can be found in the work by Zhu et al. (2022). Early works such as DIAL (Foerster et al., 2016), RIAL (Foerster et al., 2016) and CommNet (Sukhbaatar et al., 2016) introduced the concept of learning to communicate in MARL with fully connected structures, ideal for environments requiring collective decision-making. A few years later, ATOC (Jiang and Lu, 2018) and IC3Net (Singh et al., 2018) introduced gate mechanisms, enabling selective communication based on agents' local context, enhancing efficiency in scenarios where continuous information exchange is unnecessary. TarMAC (Das et al., 2020) leverages an attention mechanism for targeted communication, ensuring relevance and efficiency in message exchange. SchedNet (Wang et al., 2020a) and GA-Comm (Liu et al., 2019) utilise communication graphs, with SchedNet focusing on agent importance to manage bandwidth and GA-Comm

incorporating attention mechanisms for refined communication processes. MAGIC (Niu et al., 2021), also adopting a communication graph, adapts dynamically to changing agent states and environmental conditions.

3.3.2 Machine Theory of Mind

The concept of Machine Theory of Mind (MToM) has been explored in multi-agent systems, focusing on interpreting behaviours and enhancing cooperation. It refers to the ability of artificial agents to infer other agents' mental states, intentions, and goals. This concept, inspired by human cognitive abilities, is crucial for enhancing cooperation and coordination in multi-agent systems. Rabinowitz et al. (2018) introduced ToMNet, a meta-learning method to predict agent behaviours based on local observations. Shu and Tian (2019) proposed a hierarchical approach for a manager agent to coordinate worker agents by inferring their preferences. Shu et al. (2021) presented AGENT, a benchmark to assess machine agents' understanding of intuitive psychology. Sclar et al. (2022) created an environment requiring agents to model others' mental states for high rewards. Wang et al. (2022) proposed ToM2C, a method to decide when and with whom to communicate in multi-agent environments by predicting others' observations and goals. These recent advancements highlight the potential of MToM to significantly improve multi-agent collaboration.

3.3.3 Collaborative Perception in Autonomous Driving

Collaborative perception is a crucial aspect of autonomous driving systems, enabling vehicles to share visual and sensory data of their surroundings through Vehicle-to-Vehicle (V2V) communication. This capability helps to prevent potential accidents and improve traffic efficiency. For instance, consider the challenging scenario depicted in Figure 3.3, where a vehicle runs a red light while another vehicle approaches the intersection at high speed. In such cases, other networked vehicles could potentially alert drivers of hidden or occluded vehicles within their perception radius. However, sharing perception data can be computationally expensive and is

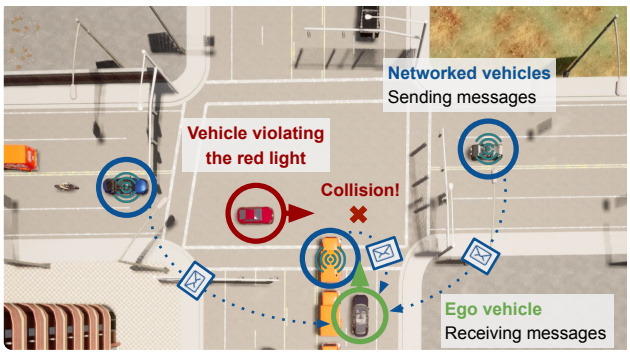


Figure 3.3: Challenging occluded scenario at an intersection. A vehicle runs a red light while another occluded vehicle approaches the intersection at high speed. Sharing messages through a V2V network can ensure safe navigation. (Adapted from Cui et al. (2022)).

also subject to malicious attacks, packet loss, or communication delays.

Collaborative perception methods can be categorised based on their data treatment strategy, which can be early fusion, intermediate or deep fusion, or late fusion. Early fusion methods communicate raw sensor data to neighbouring vehicles, but this data can be difficult to handle in real time due to its high dimensionality. Intermediate fusion methods fuse features extracted from raw data, while late fusion methods communicate perception results after the post-processing stage.

Recent studies in this field have explored intermediate fusion collaborative perception pipelines, which have shown promising results. For example, Xu et al. (2022a) presented a method that combines LiDAR data from nearby vehicles and applies a Vision Transformer for box regression and classification. Similarly, Wang et al. (2020b) proposed a V2V communication method that aggregates LiDAR information into a compressed intermediate representation and achieves higher accuracy than other V2V communication baselines.

In addition, Xu et al. (2022b) compared different collaborative V2V perception methods over a new dataset called OpenV2V and found that intermediate fusion methods have a good balance between prediction accuracy and transmission bandwidth. Furthermore, the authors proposed a novel intermediate fusion method that uses a self-attention mechanism and outperforms all other fusion methods.

One of the main challenges of collaborative perception algorithms is algorithm scalability, par-

ticularly in settings with many agents. Various approaches have been proposed to address this challenge, including Who2Com (Liu et al., 2020b) and When2Com (Liu et al., 2020a), which determine which agent to request/send data using a handshake mechanism and construct efficient communication groups to reduce the overall bandwidth load. Additionally, Aoki et al. (2020) proposed a method that uses DRL to automatically select which information to transmit to nearby CAVs to enhance sensing capabilities.

While most studies on autonomous collaborative systems focus on perception or perception and prediction (Xu et al., 2022b; Chen et al., 2019; Xu et al., 2022a; Liu et al., 2020a,b; Rawashdeh and Wang, 2018), few studies have addressed collaborative control through V2V communication. One recent study developed a method that uses collaborative perception to determine the best control decision of the ego vehicle (Cui et al., 2022). However, this method requires behaviour cloning, which is subject to biases in the driving dataset, causal confusion, and high variance (Codevilla et al., 2019). Additionally, acquiring expert data may not always be feasible or readily available, particularly for certain edge cases.

3.3.4 Adversarial Attacks on Deep Learning and Communication

Adversarial attacks on deep learning models, initially recognised in image classification (Szegedy et al., 2014; Goodfellow et al., 2015; Nguyen et al., 2015), have since extended into areas like MARL (Figura et al., 2021; Kazari et al., 2023) and communication (Tu et al., 2021; Xue et al., 2022; Blumenkamp and Prorok, 2020). These attacks can be categorised into white box (Papernot et al., 2017), where attackers have full knowledge of the network, and black box (Brunner et al., 2019; Tu et al., 2021), executed without detailed network information. Despite extensive exploration, the specific targeting of multi-agent communication within MARL remains under-investigated.

Recent studies have made significant strides in understanding adversarial communication within MARL. Mitchell et al. (2020) introduced a method to evaluate message reliability using a modified Graph Neural Network (GNN) layer in attention-based MARL systems. Blumenkamp and Prorok (2020) discovered that agents naturally develop adversarial communication strategies

in competitive-cooperative environments. In contrast, Xue et al. (2022) studied deliberate adversarial communication and proposed a defence mechanism conceptualising it as a two-player zero-sum game. The work of Tu et al. (2021) highlights vulnerabilities in distributed deep learning, showing adversarial training's effectiveness when the threat model is known. Fung et al. (2022) proposed a decentralised trust mechanism for MARL, focused on simplified binary consensus tasks.

These studies underscore the intricate nature of adversarial strategies in MARL and the growing importance of developing robust defences. While much of the literature has focused on generating or detecting adversarial communication, defence mechanisms remain relatively underexplored, presenting a critical gap in the field.

3.3.5 Adversarial Attacks on V2X communications

Adversarial attacks on V2X communications represent a relatively new field with limited research. Notable work includes the study by Tu et al. (2021), which investigated adversarial transfer attacks on collaborative perception based on LiDAR measurements. Their findings demonstrated that even simple perturbations could result in potent attacks and that adversarial training is an effective defence mechanism when the attack model is known. Li et al. (2023b) proposed an intelligent sampling method designed to aid CAVs in achieving consensus in the presence of V2V attackers. This method involves sampling a subset of messages at each step and verifying consensus between two object detections. Similarly, Xiang et al. (2023) developed a mechanism capable of generating perturbations in LiDAR-based V2X perception systems and demonstrated that training on these perturbed scenes can significantly enhance system performance. These studies underscore the importance of developing robust defences and innovative methods to mitigate the impact of adversarial attacks on V2X communications.

3.4 Autonomous Fleet Management

Autonomous Fleet Management plays a critical role in the strategic decision-making level of autonomous vehicles, especially for large-scale operations such as Autonomous Mobility-on-Demand (AMoD) and Autonomous Ridesharing services. The coordination of thousands of autonomous vehicles in dynamic, real-time environments demands efficient and scalable control methods to maintain optimal performance and service quality. As the complexity of these systems increases, the development of sophisticated fleet management strategies becomes essential.

Research in autonomous fleet management has evolved significantly. Early studies relied on centralised approaches like fluid models and queuing theory to approximate optimal fleet operations. However, recent advancements in DRL and MARL present a new frontier for improving the efficiency, scalability, and adaptability of large-scale mobility systems. This section reviews both traditional methods and cutting-edge DRL approaches for autonomous fleet management.

3.4.1 Traditional Fleet Control Methods

Early research predominantly centred on centralised methods. Notably, Pavone et al. (2012) utilised a fluid model to approximate AMoD rebalancing, obtaining the optimal policy through Linear Programming. Building upon this foundation, Zhang and Pavone (2014) introduced a queuing-theoretical model for AMoD rebalancing based on a Jackson network. Furthermore, Braverman et al. (2019) employed queuing network models for e-hailing vehicle rebalancing and conducted a comprehensive analysis of optimal policy design. Shifting the perspective, Miao et al. (2016) employed Receding Horizon Control for real-time data-based taxi rebalancing. In the context of ridesharing, Alonso-Mora et al. (2017) developed a real-time mathematical framework addressing ride-sharing dispatching and rebalancing. Noteworthy among centralised approaches, Wallar et al. (2018) introduced rebalancing via idle vehicle assignment, significantly improving various metrics compared to the study by Alonso-Mora et al. (2017).

Decentralised methods have also been explored. For instance, Cubillos et al. (2004) proposed mixed centralised-decentralised negotiation-based fleet control. Belakaria et al. (2017) used

fog computing for decentralised vehicle dispatching and charging optimisation. Additionally, Albert et al. (2019) presented a distributed approach averaging system imbalance to enhance MoD system stability.

3.4.2 Deep Reinforcement Learning

The study of DRL techniques for Mobility-on-Demand or Ridesharing systems has received increasing attention in the last five years. A comprehensive survey was presented by Qin et al. (2022). Studies in this field can be organised in terms of their modelling approach: fully centralised single agent, independent learning and fully observable critic. In the fully centralised approach, a single agent takes all actions based on the joint observations. In contrast, the independent learning approach allows agents to undertake actions based on their own observations. In the fully observable critic approach, the agents are also independent but are helped by an omniscient critic.

Fully Centralised Single Agent

The first DRL approach to fleet control involves training a centralised controller—a single RL agent—to determine the joint actions of all vehicles at each time step. While conceptually simple, this approach faces scalability challenges, as the state and action spaces grow exponentially with the number of vehicles, making the problem computationally intractable (OroojlooyJadid and Hajinezhad, 2020). Despite these limitations, several studies have successfully applied centralised controllers to model MoD problems.

For instance, Kullman et al. (2022) proposed a centralised e-hailing operator responsible for managing a fleet of electric vehicles. In this system, the controller determines both the assignment of vehicles to customer requests and the repositioning or recharging actions. Similarly, Mao et al. (2020) developed an actor-critic framework for AMoD systems, where a central dispatcher optimises vehicle repositioning by determining the number of available vehicles to relocate across different geographic zones.

Deng et al. (2020) addressed the vehicle rebalancing problem through a Semi-Markov Decision Problem (SMDP), applying Proximal Policy Iteration (PPO) to manage the redistribution of vehicles across a network. The centralised controller observes the system's state and takes rebalancing actions to ensure vehicles are directed to nodes where demand is high. In the context of taxi carpooling, Jindal et al. (2018) developed a centralised decision-making framework, combining a deep neural network for trip time prediction with a tabular Q-learning algorithm to optimise carpooling efficiency.

Additionally, Chen et al. (2021) explored centralised control for spatial-temporal pricing in ride-sourcing platforms using a PPO framework. Their model optimised both spatial and dynamic pricing strategies, significantly improving platform profit while better balancing supply and demand.

Independent Learning

The second approach to address the scalability issue in DRL is Independent Learning, where each agent is trained to make decisions based solely on its own observation. The Independent Q-learning (IQL) algorithm, a natural extension of the Q-learning algorithm to the multi-agent setting, is the most common form of Independent Learning. In IQL, agents do not have any knowledge of the other agents and treat them as part of the environment, allowing them to learn their value function independently and avoiding the scalability issues of single-agent RL.

A recent extension of the IQL algorithm uses Deep Neural Networks to approximate each agent's value function. This method is called Independent Deep Q-Networks (IDQN) (Tampuu et al., 2015), and has been used extensively for dynamic fleet management in the last ten years (Oda and Tachibana, 2018; Wen et al., 2017; Oda and Joe-Wong, 2018; Alabbasi et al., 2019; Gueriau and Dusparic, 2018; Gueriau et al., 2020; Holler et al., 2019).

IDQN has been shown to achieve higher-performing solutions than traditional fleet management methods. Wen et al. (2017) used the IDQN algorithm for MoD system rebalancing based on agents' local observations and without explicit agent coordination. They obtained better

average waiting times when compared to a Simple Anticipatory Rebalancing (SAR) policy. Oda and Joe-Wong (2018) compared the performance of the IDQN algorithm with a centralised Receding-Horizon Control (RHC) policy and showed that using IDQN, the number of unserved requests is 20 % lower. Alabbasi et al. (2019) also showed that an IDQN-based policy could outperform RHC and other simple heuristics.

Other articles have focused on adding different problem features to improve IDQN performance. For example, Alabbasi et al. (2019) added the predicted number of vehicles and future demand to the state variable, while Gueriau and Dusparic (2018) addressed the AMoD joint problem of dispatching and rebalancing using IQL. Additionally, Gueriau et al. (2020) extended the previous model to consider traffic congestion by integrating a traffic micro-simulator to represent traffic conditions and calculate vehicle speed and position for each time step.

Fully observable critic

The third approach to fleet control problems involves training a centralised controller that collects information from other agents, but lets agents individually learn their own policy or value function. This method is known as fully observable critic (OroojlooyJadid and Hajinezhad, 2020). The benefit behind this idea is that the centralised controller—the critic—observes the actions and local observations of all agents, and therefore the overall environment can become stationary.

A popular application of the fully observable critic is the Centralised Learning and Decentralised Execution (CLDE) approach (Foerster et al., 2017; Lowe et al., 2020). Under this approach, individual agent policies are trained using the gradient estimated by a fully observable critic during the training phase. During execution, each agent takes actions based exclusively on their trained policy, which frequently is a policy neural network.

The CLDE paradigm has become increasingly popular in MARL-related problems, and a few articles regarding fleet control have been published. de Lima et al. (2020) proposed a method for efficient e-hailing vehicle dispatching based on QMIX—a value-based algorithm that works

under the CLDE paradigm (Rashid et al., 2018a). They show that their method outperformed IDQN and completed all rides faster for a fixed number of vehicles and passengers. Further, Lin et al. (2018) compared the performance of various algorithms using a large-scale e-haling simulator calibrated with real data provided by the company Didi Chuxing. Among other baselines, they compared the IDQN algorithm to a proposed contextual DQN (cDQN) and a contextual Actor-Critic (cA2C), which use context information, such as geographical context, to guarantee agent coordination. While both algorithms achieve agent coordination, cA2C is more flexible and efficient due to its CLDE architecture.

Promising results have been published in small-scale scenarios under the CLDE paradigm. However, the critic still suffers from the curse of dimensionality because it concatenates agents' observations (OroojlooyJadid and Hajinezhad, 2020). To tackle this, Yang et al. (2018) proposed a Mean Field Theory (MF)-based approach, dealing with large state spaces observed by the critic. Recent studies have employed MF in MoD problems. Li et al. (2019) used MF for vehicle dispatching, achieving improved performance in comparison to optimisation methods and heuristics. Shou and Di (2020) coupled an MF-based approach with MARL for multi-driver repositioning tasks, ensuring a better system equilibrium by considering a higher-level reward design methodology.

A summary of the different DRL approaches in the reviewed literature is presented in Table 3.4. There is a need to compare these modelling perspectives in terms of performance, contrasting the different learning curves and the final reward and fleet management metrics. Moreover, the applicability of these methods to large-scale and potentially non-stationary environments should also be assessed.

3.5 Sim-to-real Transfer for Autonomous Driving

Sim-to-real refers to the process of transferring knowledge or skills acquired in a simulated environment to the real world, addressing the challenges posed by discrepancies between the two, known as the *reality gap*. To bridge this gap, various domain-adaptation approaches have

Table 3.4: Different DRL modelling approaches for dynamic fleet control.

Modelling approach	Description	Articles
Centralised single-agent	• Single RL agent that observes the complete state space. • Environment is stationary. • State and action space can grow exponentially.	(Kullman et al., 2022; Mao et al., 2020), (Deng et al., 2020; Jindal et al., 2018), (Xie et al., 2023)
Independent learning	• Each vehicle is represented as an independent RL agent. • More scalable than the single-agent approach. • Environment can become non-stationary.	(Oda and Joe-Wong, 2018), (Gueriau and Dusparic, 2018; Gueriau et al., 2020) (Alabbasi et al., 2019)
Fully observable critic	• Each vehicle is an RL agent with access to centralised information. • Non-stationarity can potentially be overcome. • Centralised state can grow exponentially with the number of vehicles	(de Lima et al., 2020; Rashid et al., 2018a), (Lin et al., 2018; Li et al., 2019), (Shou and Di, 2020; Xie et al., 2023)

been developed (Zhao et al., 2020). The area with the highest contributions in sim-to-real is *robotic manipulation*. Methods used in this field include imitation learning, data augmentation, and real-world RL. The latter is particularly challenging to replicate due to limited resources, safety concerns, and the difficulty of resetting training runs. Although there are straightforward methods to automate resets in real-world robotic manipulation (Eysenbach et al., 2017) or to continue training efficiently without resetting (Zhu et al., 2020a; Gupta et al., 2021), these techniques are not easily transferable to AVs. In this domain, human intervention is necessary to reset the environment or prevent catastrophic events. Consequently, most techniques used in robotic manipulation cannot be adapted to AV environments.

In the context of addressing the *reality gap* for autonomous driving, various frameworks and infrastructures have been proposed to enhance the training and robustness of these systems. Wang et al. (2021a) proposed ModEL, a modular infrastructure which considered perception, planning, and control for AVs, each being trained using RL. They used vision as the agent's main sensory perception and the CARLA simulator (Dosovitskiy et al., 2017) during training for data augmentation and domain generalisation in order to improve overall agent robustness. So et al. (2022) explored how sim-to-real techniques from smaller-scale robotic domains can be applied to off-road autonomous driving. They proposed Sim2Seg, the first method to bridge the visual *sim-to-real gap* in autonomous driving. Wu et al. (2023) utilised domain augmentation techniques and allowed for intermittent human guidance to enhance training efficiency and improve real-world performance in autonomous navigation tasks. This last approach is presented in figure 3.4.

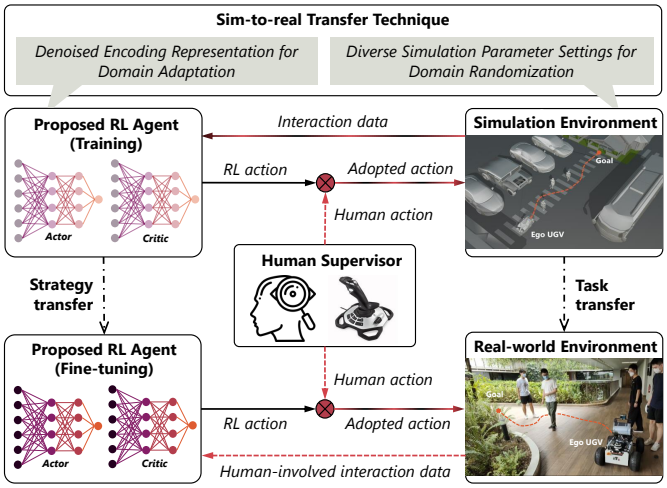


Figure 3.4: Sim-to-real transfer for Autonomous Driving with human guidance. (Adapted from Wu et al. (2023)).

Furthermore, in the case of multi-agent systems, there are more *reality gaps* compared to single-agent settings. Three significant gaps have been reported in Suh et al. (2019): *the control architecture* gap, which relates to the tendency of simulators to synchronise the actions of all agents at each time step; *the observation* gap, which relates to the limited perception of agents in scaled-out environments; and *the communication* gap which, similar to the previous one, relates to the highly limited and inconsistent communication within multi-agent systems. Given traditional methods of software simulation, the aforementioned gaps are non-trivial to overcome, even with a redesign of the simulation itself. The authors (Suh et al., 2019) proposed a method called Agent Decentralised Organization (ADO), which encourages agents to share a board of information provided at certain time frames without the necessity for the information to be complete or up to date.

Although research in Sim-to-real for AV learning is extensive, the main focus of the literature is generalisability over diverse environments, not different actors. This can make progress in the literature slower since the hardware is usually significantly different and not open source. Thankfully, there are attempts to standardise the research hardware with open source sets like Duckietown (Paull et al., 2017), and DeepRacer (Balaji et al., 2019b), which ensure that more prior research becomes easier to reproduce consistently. However, no study has focused on transferring multi-agent policies for autonomous driving to the real world.

3.6 Summary of gaps in the current literature

This literature review has covered key areas of research essential for the development of AV systems, with a particular emphasis on multi-agent learning methods. The following gaps have been identified:

- We examined the optimisation of traffic flow and the integration of risk-aware decision-making processes, which are critical for ensuring efficient and safe operation of AV systems. The review highlighted the importance of both heuristic methods and MARL in managing complex urban traffic scenarios. Despite these efforts, safety and coordination in dynamic urban environments remain unresolved issues in MARL for autonomous driving, underscoring the need for innovative solutions to address the unpredictability and complexity of these scenarios. While TTC-based and other heuristics are easy to apply in simplified scenarios, they fall short in settings that require high levels of coordination, such as manoeuvring in the presence of emergency vehicles.

- The literature on communicative MARL and collaborative perception for AV coordination shows that integrating these mechanisms enhances CAVs' ability to share environmental knowledge and coordinate actions, especially in low visibility scenarios. However, few studies address the problem from an end-to-end perspective——encompassing perception to control. Most research focuses on perception or perception and prediction, often relying on behaviour cloning, which is subject to biases, causal confusion, and high variance. Additionally, expert data is not always feasible or readily available, particularly for certain edge cases.

- Collaborative V2X methods are subject to adversarial attacks, underscoring the importance of developing robust defences and innovative methods to mitigate their impact on V2X communications. These studies highlight the complexity of adversarial strategies in MARL and the critical need for effective defence mechanisms. While most of the literature focuses on generating or detecting adversarial communication, there are few works addressing robust defence mechanisms.

- Large-scale AV fleet management has recently been a significant focus of research, highlighting the need for efficient methods that can scale to city-sized networks. While MARL methods have shown promise in fleet management, their evaluation has primarily focused on small-scale scenarios and simplified networks, leaving a significant gap in scaling these methods to larger, more complex systems. Decentralised learning methods, though potentially efficient, face challenges related to non-stationarity due to agents learning concurrently in the same environment. Therefore, new methods must be developed to tackle non-stationarity and ensure stable learning.

- Research on Sim-to-real primarily focuses on robotic manipulation, with some exploration in single-agent navigation. However, the *reality gap* becomes significantly larger for multi-agent systems, such as those involving multiple AVs. These systems introduce additional complexities, including agent collaboration, environment synchronisation, and limited agent perception. Little work has been done to bridge the gap between simulation and reality in multi-AV systems. Although attempts have been made to make MARL more robust to uncertainty, these methods have not yet been tested in real-world scenarios.

The following chapters will elaborate on the research questions derived from these gaps and outline the methodologies to be employed, ultimately contributing to the field of autonomous vehicle navigation and coordination.

Chapter 4

Ensuring Coordination and Safety in Real-World MARL through Sim-to-Real Transfer

This chapter has been adapted from the following publication:

E. Candela, L. Parada, L. Marques, T.-A. Georgescu, Y. Demiris, and P. Angeloudis. Transferring Multi-Agent Reinforcement Learning Policies for Autonomous Driving using Sim-to-Real. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8814–8820, Oct. 2022. ISSN: 2153-0866

4.1 Introduction

This chapter marks the beginning of the methodological contributions of this thesis, addressing the challenge of ensuring coordination and safety in dynamic, real-world environments. It focuses on operational-level decision-making, where precise, real-time control is essential for transferring simulation-trained policies to unpredictable real-world conditions, in accordance with the multi-level framework introduced in Chapter 1 (see Figure 4.1).

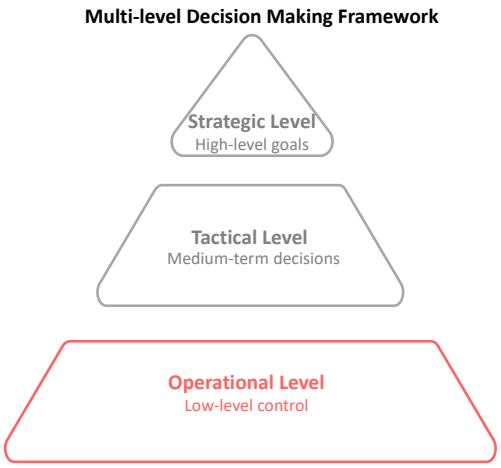


Figure 4.1: Sim-to-real MARL framework at the operational level.

As highlighted in the Literature Review chapter, a key challenge for Multi-Agent Reinforcement Learning (MARL) in autonomous driving is bridging the *reality gap*—the discrepancy between simulation performance and real-world deployment (Tobin et al., 2017). While simulations offer a controlled environment for training MARL algorithms, they often fail to account for real-world complexities such as sensor inaccuracies and environmental variability. These challenges are especially critical for operational tasks like real-time obstacle avoidance and trajectory planning.

Several methods have been proposed to bridge the *reality gap* in single-agent tasks, particularly in robotic manipulation, including System Identification (Kaspar et al., 2020), Domain Randomisation (Matas et al., 2018), and Domain Adaptation (Bassani et al., 2020; Zhao et al., 2020; Eysenbach et al., 2017). Despite these efforts, closing the *reality gap* in single-agent applications continues to pose significant challenges.

The *reality gap* can become even larger for multi-agent systems, such as those involving AVs. These systems introduce additional complexities to the sim-to-real problem, such as agent coordination, environment synchronisation and limited agent perception. Little work has been done on bridging the gap between simulation and reality in multi-AV systems. Attempts have been made to make MARL more robust to uncertainty (Suh et al., 2019; Zhang et al., 2020), although these methods have not been tested in the real world.

In this chapter, we present a method to train multi-agent autonomous driving policies in simulation and effectively transfer them to reality. For this purpose, we present a multi-agent autonomous driving gym environment that resembles the Duckietown testbed (Paull et al., 2017), which consists of small differential drive robots with two conventional wheels and a

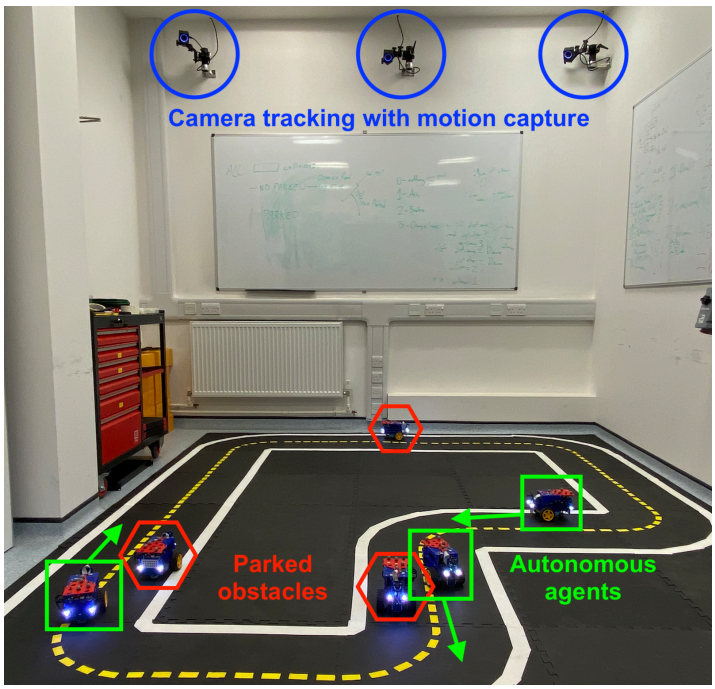


Figure 4.2: Sim2Real Setup: The goal is to train autonomous vehicles to swiftly navigate the track, avoiding collisions with obstacles. Using Multi-Agent Deep Reinforcement Learning with domain randomisation, we reduce the *sim-to-real gap* by introducing noise and uncertainty in simulations. The refined policies are then applied to a fleet of Duckiebot robots, tracked by a motion capture camera system.

passive caster wheel. We then train a wide range of multi-agent policies in simulation using MARL and domain randomisation (Tobin et al., 2017), and then transfer them to the miniature autonomous vehicles. We compare this method to a traditional rule-based approach, involving scenarios with up to three agents and three stationary vehicles serving as obstacles. A summary of the sim-to-real setup is presented in figure 4.2. Through extensive experiments, we demonstrate the advantages of training policies with domain randomisation for more effective real-world transfer.

4.2 Framework

This section presents our framework for enabling effective communication and collaboration among autonomous vehicles in the Duckietown Multi-Agent Autonomous Driving Environment (Duckie-MAAD). Utilising Multi-Agent Proximal Policy Optimization (MAPPO), the framework integrates simulation with real-world testing, employing domain randomisation to

effectively bridge the sim-to-real gap. It details the architecture and operational specifics essential for robust multi-agent system performance in dynamic environments.

4.2.1 MAPPO-based architecture

MAPPO (Yu et al., 2021) is an extension of the Proximal Policy Optimization algorithm to the multi-agent setting. As an on-policy method, it can be less sample efficient than off-policy methods such as MADDPG (Lowe et al., 2017) and QMIX (Rashid et al., 2018b). Despite this fact, MAPPO has been found to be a strong method for cooperative environments, outperforming state-of-the-art off-policy methods and achieving similar sample efficiency in practice. In addition, knowledge of prior states gets recalled through the usage of recurrent neural networks (RNNs) for the actor and critic networks.

The architecture of MAPPO consists of two networks: an actor network that uses each agent's local observation o_i^t, and a critic network that has access to the global state s^t. This design aligns with the Centralised Training Decentralised Execution (CTDE) paradigm. During training, the centralised critic network optimises cooperation by computing a shared advantage function $A(s^t, a^t)$ for all agents based on joint observations and rewards. However, during execution, each agent's actor network selects actions $a_i^t \sim \pi_\theta(a_i|o_i^t)$ using only local observations, ensuring scalability and decentralisation. For simplicity, we assume parameter sharing among agents, though MAPPO allows for distinct networks per agent. A more detailed explanation of the MAPPO algorithm, along with pseudocode, can be found in Appendix B.

4.2.2 Duckietown testbed

Duckietown (Paull et al., 2017) is a multi-robot testbed that consists of several small autonomous robots (Duckiebots) that transit inside a city (Duckietown). Each Duckiebot is a differential drive vehicle equipped with wheel encoders, a monocular camera, an Inertial Measurement Unit (IMU) and a time of flight sensor.

For positioning, we use the NaturalPoint OptiTrack MoCap system (OptiTrack, 2022) to obtain the real-time pose of the Duckiebots. A unique asymmetric passive marker configuration is used for each car, and the system was calibrated to be accurate up to 0.5 mm. The 3D pose of each car is broadcasted to the simulator through the NatNet SDK, such that position (x, y) and orientation (θ) are directly obtained after a coordinate transformation, and velocities are calculated from applying finite differences to consecutive measurements.

4.2.3 Duckietown Multi-Agent Autonomous Driving Environment

For the purpose of training and evaluating multi-AV policies, we introduce a new environment called Duckietown Multi-Agent Autonomous Driving Environment (Duckie-MAAD), which is based on the Gym-Duckietown environment (Chevalier et al., 2018). The original Gym-Duckietown is a single-agent lane-following environment that can be used to test different RL methods. We extend this environment to the multi-agent setting and allow agents to make decisions based on local observations. The agents (Duckiebots) can move around a track, making decisions to accelerate, brake, or steer. In addition, each agent can perceive nearby agents and objects in the track within a given radius. Based on perception, proximity and collision penalties can be assigned to agents, as well as a penalty for leaving the track. Although the agents follow a predefined path by waypoints, they have to learn to stay inside the track, because going too fast will take them off. Figure 4.3 shows a test track with 3 agents (the moving cars) for the Duckie-MAAD environment.

Upon a new environment step, the MAPPO algorithm takes in the observations of the agents and outputs action a_i for each car. Using this as input, a Path Following logic is implemented that determines for each car its next goal waypoint (within a pre-defined list that extends along each lane), and calculates the required linear and angular velocities (v, ω) to reach it. These are passed to the differential drive Inverse Kinematics models, which produce the left and right wheel velocities (V_l, V_r) that lead to this motion. Said velocities are then either fed to the simulated vehicles, where the pose is updated via a Non-linear Dynamics model (Ercan, 2019); or to the real cars, where the pose is then updated via the OptiTrack MoCap system. The new

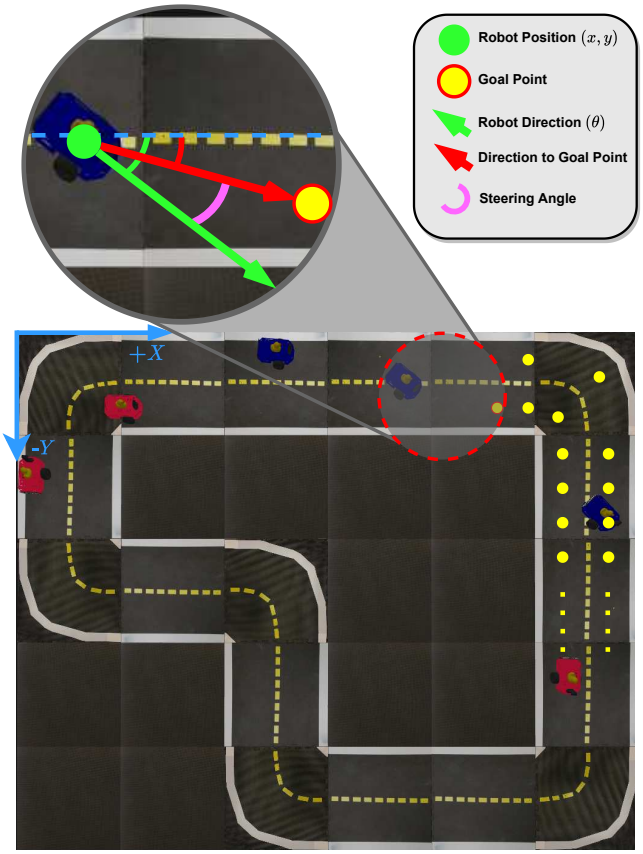


Figure 4.3: Test track, used in simulation and reality, for the Duckie-MAAD gym environment. Yellow circles represent the waypoints along each lane, which extend all the way around the loop. The *goal point* is determined by the Path Following function and the steering angle required to reach it is the difference between the car's direction in world coordinates and the angle between the car's position and the *goal point*.

robot poses are then used to calculate the Reward R and the end of the step is reached. A detailed diagram illustrating this step update can be found in Figure 4.4.

We define the individual elements of the Duckietown Multi-Agent Autonomous Driving Environment as follows:

- Agents N: Each Duckiebot is treated as an individual agent. All agents are independent, have a local observation of the environment and obtain a unique reward.
- Observation Ω: The local observation of each Duckiebot is composed of the steering angle, the distance to the centre of the lane, the angle with respect to a tangent to the current lane, local distances to the closest Duckiebots in the same and opposite lane, their respective longitudinal velocities, and a binary variable that shows whether the agent is

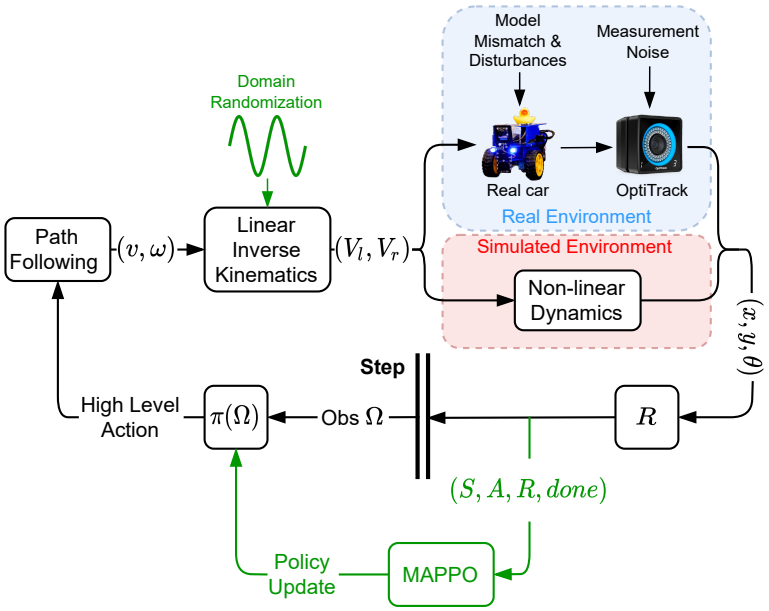


Figure 4.4: Duckie-MAAD architecture. The step update loop is as follows: 1) The agent takes action following policy π. 2) The Path Following function selects the next target waypoint and calculates the linear and angular velocities required to reach it 3) The wheel velocities that lead to such speeds are calculated using (linear) differential drive inverse Kinematics and sent to the agents (and domain randomisation is added if training MARL). 4) The resulting pose for each agent is obtained either from real life or in the simulator, together with a reward R; If training MARL, 5) the policy is updated using MAPPO. The green elements are only when training MARL.

off the track or not.

- Action A: A discrete action space with four possible decisions – accelerate (0.25 m/s^2); brake (-0.25 m/s^2); steer; keep previous velocity (no acceleration).

- Reward function R: $v - 5c - 5t - 0.5l$; where v is the agent's measure velocity (m/s), and binary variables c representing a collision, t capturing if the agent is out of the track, and l if the agent is changing lanes.

4.2.4 Simulation to reality transfer

There is a significant gap between the simulation environment and reality due to several reasons. First, the agent dynamics within the Duckie-MAAD gym environment are based on a dynamical model, which is based on various assumptions, such as a symmetrical mass distribution of the Duckiebots and that the motors operate in steady-state mode. Second, in the simulator,

agent steps are performed sequentially according to the environment's internal clock; meanwhile, in reality, the agents may take their own steps asynchronously. Third, there are certain inaccuracies related to the OptiTrack positioning system. Fourth, although all the Duckiebots share the same architecture, in practice, they all behave differently due to small differences in their individual components. All of the above create a significant sim-to-real gap that must be addressed in order to successfully implement multi-agent policies trained in simulation.

To tackle this gap, we use domain randomisation, which is a technique that consists of training a DRL model over a variety of simulated environments with randomised parameters. The objective is to train a model that can adapt to the real-world environment, which is expected to be a sample of the space of environments created through the randomisation procedure. The parameters of the Inverse Kinematics Model that showed the most variance among the physical cars were K, gain, and trim. Moreover, there was a high error in the observed steering angle during initial testing. Therefore, we chose these parameters for domain randomisation within the Duckie-MAAD training RL environment:

- Steering factor ($unitless$)
- Motor constant (K) ($Nm/\sqrt{W}$)
- Gain ($unitless$)
- Trim ($unitless$)
- Steering error (rad)

4.3 Experiments

4.3.1 Setup

To implement and validate the proposed methodology, we made use of the Duckie-MAAD environment to first train and then test various policies in simulation. Afterwards, experiments

in real life were conducted following the same goals, rewards and settings. The specific environment utilised consisted of 3 agents and 3 permanently parked cars randomly spawned at the beginning of each episode throughout the test track shown in Figure 4.3. Agents had the goal to maximise their reward by going around the track as fast as possible while avoiding leaving the bounds of the track and colliding with other cars (either parked or other moving agents). A minimum speed of 0.1 m/s was set for all agents; and a maximum speed of 0.3, 0.4 and 0.5 m/s for each agent, respectively. Both the simulations and real experiments were run at discrete time steps with a frame rate of 10 Hz. The 8 PrimeX13 cameras of the MoCap setup captured at 120 Hz, providing more than sufficient temporal resolution.

We trained three policies using the MAPPO method with the parameters presented in figure 4.1. Each policy was trained using a different level of domain randomisation (none, medium, high) with the distributions shown in Table 4.2 – Normal $N(\mu, \sigma)$, or Uniform $U(a, b)$. For training, a workstation was used with an AMD Ryzen Threadripper 3990x 64-core processor and an NVIDIA RTX 3090 GPU, which resulted in a training time of approximately 3 hours for each policy.

Table 4.1: Training Parameters for MARL Experiments

Parameter	**Value**
Number of Episodes	2000
Steps per Episode	400
Learning Rate	5×10^{-4}
Critic Learning Rate	5×10^{-4}
PPO Epochs	15
Entropy Coefficient	0.01
Actor Network	MLP(64, 64, 4)
Critic Network	MLP(64, 64, 1)

As a baseline, we implemented Gipps' lane changing model (Gipps, 1986) – which is a commonly used rule-based algorithm for describing driving behaviour – and set the safe following distances according to the RSS model introduced by (Shalev-Shwartz et al., 2017).

Table 4.2: Domain randomisation distributions.

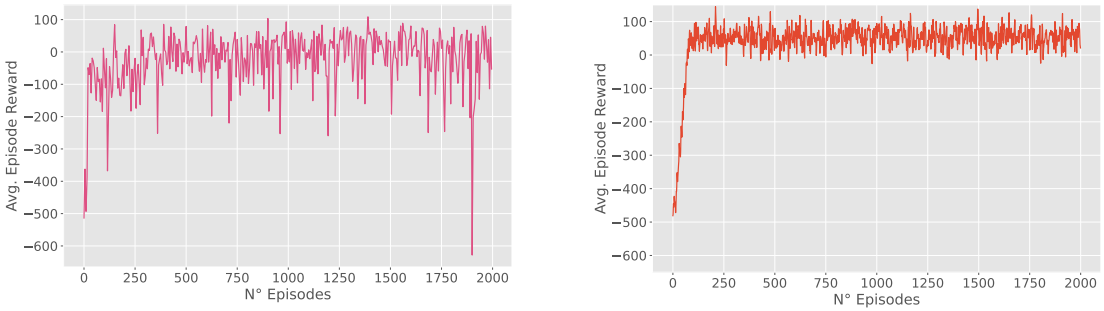
	Rule-based	**No D.R.**	**Med D.R.**	**High D.R.**
Steer factor	1	1	U(0.8,1.2)	U(0.5,1.5)
K	27	27	U(22,32)	U(14,40)
Gain	1	1	U(0.8,1.2)	U(0.5,1.5)
Trim	0	0	U(-0.1,0.1)	U(-0.15,0.15)
Steer error	0	0	N(0,0.1)	N(0,0.5)

The four policies were tested first in the simulated environment and then in the Duckietown testbed to measure and compare their performance and transferability to the real world. To achieve statistical significance and reduce the variance of the results, 30 runs were executed in the simulator and then another 30 in real life for each policy, with all cars being randomly spawned at the beginning of each run. Each run lasted 400 steps, granting enough time for the agents to go around the track at least once.

4.3.2 Analysis

The first step was training the four policies in the simulator. All converged after approximately 500 episodes, as shown in Figure 4.5, which compares the learning curves for no-domain and medium-domain randomisation. The curve on the left, with domain randomisation, shows more variability due to the added noise and environmental changes, leading to less stable convergence but improved policy robustness. In contrast, the smoother curves on the right, without noise, indicate a more stable learning process but result in policies that are less capable of generalising to real-world conditions, as discussed later.

The average rewards achieved by each policy in the simulated environment and real life are presented in Figure 4.6; the averages were computed over all scenarios and cars. It can be seen that policies clearly perform worse in reality than in simulation. This is due to the previously discussed *reality gap*, which in this case is significant mainly because of the high unreliability of the Duckiebot robots. If the simulated environment perfectly captured the real



(a) Reward convergence during training of Med D.R. policy.

(b) Reward convergence during training without D.R.

Figure 4.5: Comparison of reward convergence with and without domain randomisation (D.R.) during training.

world, then the difference between simulated and real-life rewards would be negligible, and no domain randomisation would be required. Thus, the *reality gap* was measured as the difference between the reward achieved by a policy trained in the simulator with no domain randomisation and the reward obtained by applying that same policy in real life.

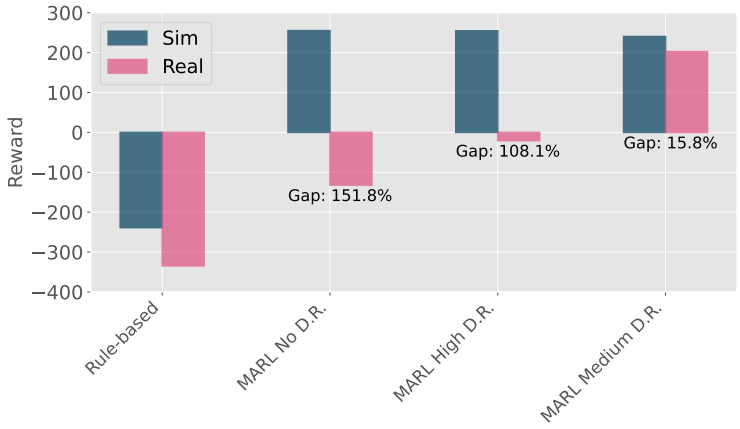


Figure 4.6: Average rewards (over scenarios and cars) for policies in simulation and real world. MARL policies clearly outperform the rule-based baseline in the simulator and in the real world. The policy with medium domain randomisation bridges the *reality gap* the best, as it reduces the gap by almost 90%.

The rule-based policy performed the worst, as was expected because despite following safety distances, it cannot adapt to other agents not precisely following their lanes. The policy trained with MARL and no domain randomisation achieved the best rewards in simulation but the worst in real life among the MARL-trained policies. Note that, despite slightly decreasing the simulated performance, the addition of domain randomisation improves rewards in real life.

The policy with a medium level of domain randomisation performed better than the one with a high level because the latter learned to be more conservative, as can be seen in the following figures. The poor performance of the untrained rule-based algorithm can be used as a proxy for the *reality gap* caused mainly by the unreliability and unpredictability of the real Duckiebots.

To better understand the resulting rewards of the policies, we analyse the four aspects that compose the reward function: speed, staying within the designated track, collisions, and lane changes. The numerical results of both the simulation and real tests are presented in Figures 4.7 and 4.8.

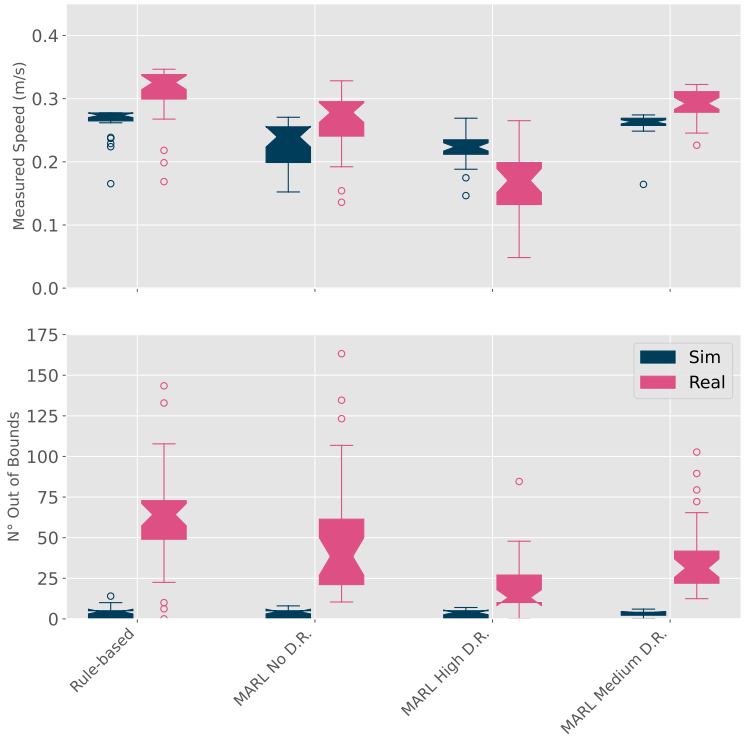


Figure 4.7: Distribution of performance metrics across different policies: 1) measured speed; 2) the number of times a vehicle left the track. The rule-based method tends to present the highest speed and track violations, closely followed by the MARL policy with no D.R. The policy obtained with medium D.R. (right-hand side policy) is the best, followed by the one with high D.R. which tends to be more conservative in terms of speed – leading to the fewest track exits, especially in real life.

The average speed is the highest for the rule-based policy, closely followed by the med D.R. one. The policy with the slowest average measured speed is the one with high D.R. – confirming that this is the most conservative policy, especially in real life. Similarly, for the times an agent leaves the track bounds, the rule-based presents the highest number and the high D.R.

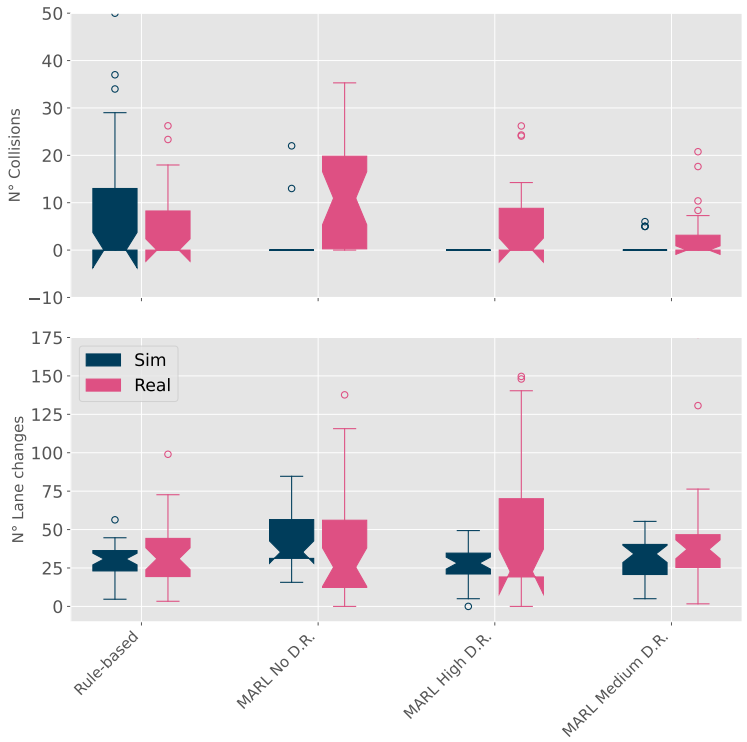


Figure 4.8: Distribution of performance metrics across different policies: 1) number of collisions; 2) number of lane changes. In general, the policies trained with D.R. perform better in real life, partly because they change lanes significantly more times (compared to simulation) to avoid hazardous situations. The medium D.R. policy achieves the least amount of collisions in real-life.

the lowest; notice that this phenomenon happens mainly in real life and not in the simulated environment. For collisions, note that for all three MARL policies, there are virtually no collisions in simulation and that the policies trained with domain randomisation tend to have lower collisions compared to the other policies. Finally, the average number of times an agent changes lanes is significantly larger in real life compared to simulation for policies trained with domain randomisation, which can be interpreted as the agents choosing to switch lanes more frequently to avoid the potential crashes that happen in real life.

4.4 Concluding Remarks

Effective coordination and collaboration among AVs have the potential to deliver significant safety and efficiency benefits across a range of industries. MARL can facilitate this process by

allowing agents to learn how to work together effectively through the sharing of observations and rewards. However, for MARL to be successfully applied, it is crucial to develop robust methods that can effectively bridge the reality gap between simulated training environments and real-world settings.

In this chapter, we presented a method to train policies using MARL and to reduce the *reality gap* when transferring them to the real world via adding domain randomisation during training, which we show has a significant and positive impact on real performance compared to rule-based methods or policies trained without different levels of domain randomisation.

It is important to mention that despite the performance improvements observed when using domain randomisation, its use presents diminishing returns, as seen with the overly conservative policy, for it cannot completely close the *reality gap* without increasing the fidelity of the simulator. Additionally, the amount of domain randomisation to be used is case-specific, and a theory for the selection of domain randomisation remains an open question. The quantification and description of *reality gaps* presents another opportunity for future research.

Chapter 5

Coordinated Manoeuvring in Presence of Emergency Vehicles

This chapter has been adapted from the following publication: *L. Parada, E. Candela, L. Marques, and P. Angeloudis. Safe and efficient manoeuvring for emergency vehicles in autonomous traffic using multi-agent proximal policy optimisation. Transportmetrica A: Transport Science, 0(0):1–29, 2023.*

5.1 Introduction

While the previous chapter addressed the operational challenge of sim-to-real transfer for Autonomous Vehicle (AV) systems in a controlled setting, this chapter shifts to more intricate urban scenarios that require tactical-level decision-making (see Figure 5.1). Tactical decisions focus on manoeuvre planning, such as lane changes, merging, and overtaking, which are essential for navigating dynamic and unpredictable urban traffic.

As highlighted in Chapter 1, one of the major challenges in autonomous driving is managing *edge cases*—rare but high-risk scenarios requiring precise responses. Among these, manoeuvring in the presence of emergency vehicles (EMVs) is particularly challenging due to the urgency and

unpredictability involved. In such situations, AVs must make tactical decisions that quickly adapt to emergency vehicles while ensuring smooth traffic flow and overall safety.

Although Multi-Agent Reinforcement Learning (MARL) has shown promise in improving coordination among AVs, dealing with coordination in scenarios involving EMVs introduces additional complexity. These situations require quick and effective tactical-level decisions to ensure the safe and efficient manoeuvring. This chapter presents a MARL-based coordination method specifically designed to handle manoeuvres around EMVs, contributing to the overall objective of ensuring safety and reliability in high-risk urban scenarios.

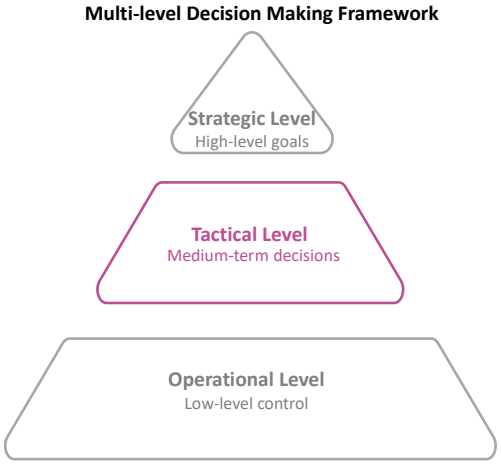


Figure 5.1: This chapter focuses on tactical-level decision-making in urban environments, specifically manoeuvring around emergency vehicles.

The unpredictable movements of EMVs, such as high-speed driving and contraflow in certain countries like the UK, contribute to their involvement in numerous, sometimes fatal, accidents. In 2020, the United States recorded 180 deaths from collisions involving emergency vehicles (EMVs). These incidents are becoming increasingly concerning due to rising road congestion (NSC, 2020). In the UK, 2019 saw a 13-year high in police-related road incidents (Gayle, 2019), underscoring the significant financial and safety impacts of such accidents.

Current vehicle autonomy systems, especially those at SAE Automation Level 2 (SAE, 2019), face difficulties in navigating around EMVs, evidenced by recent incidents involving autonomous vehicles obstructing emergency responses and colliding with EMVs (Marshall, 2022; Isidore and Valdes-Dapena, 2021). This highlights the necessity for algorithms capable of safe EMV interaction.

Research on EMV manoeuvring within autonomous traffic has predominantly focused on vehicle sequence optimisation to minimise EMV delay, employing methods ranging from genetic

algorithms to game-theoretic control (Humayun et al., 2022; Nellore and Hancke, 2016; Lu and Kim, 2017; Buckman et al., 2021). These studies, however, often rely on simplified traffic assumptions, highlighting a gap in adaptable solutions for dynamic road scenarios.

This chapter introduces a MARL approach that combines Multi-Agent Proximal Policy Optimisation (MAPPO) and a collision risk model for safe manoeuvring around EMVs. Our method is empirically validated against a Model Predictive Control (MPC) (Zhou et al., 2020a) baseline across various scenarios to demonstrate its efficacy in enhancing traffic safety and efficiency. Moreover, we conduct a comprehensive evaluation of this method in a wide range of scenarios, including assessing the trade-offs between traffic efficiency and safety, scaling the approach to several AVs, analysing different distributions of mixed human and autonomous traffic, and examining the various levels of cooperation and competition among agents.

We argue that a MARL-based solution offers superior adaptability and safety outcomes in challenging urban contexts compared to traditional rule-based approaches. This hypothesis is supported by the adaptability of learning-based methods, the multi-agent nature of driving, and the capability of machine learning algorithms to design safety-focused rewards. By conceptualising EMV overtaking as a stochastic game, our approach facilitates independent, informed decision-making by agents, aiming to improve overall traffic dynamics and safety in the presence of EMVs.

The rest of the chapter is organised as follows. Section 5.2 contains the risk modelling approach. Section 5.3 contains the MARL formulation of the problem, along with the Emergency Vehicle Gym environment. Section 5.4 presents the main results of the different case studies. Finally, section 5.5 presents the conclusion along with insights into future work.

5.2 Modelling risk

The deployment of AVs on roads poses significant safety risks, especially in complex hazardous scenarios involving humans, which are essential for training and validating AV controllers. This necessitates risk-aware AV controllers that can better predict and avoid dangerous situations.

The behaviour of drivers in situations involving EMVs differs significantly from normal driving scenarios. EMVs must perform riskier manoeuvres and cannot be overly conservative, while other road users are expected to yield to them. Therefore, traditional collision modelling should be adapted to better suit EMV scenarios. Our focus is on collision avoidability rather than collision probability, as EMVs must drive more aggressively without causing situations that lead to unavoidable collisions.

We introduce a single risk index $r \in [0, 1]$ that captures the avoidability of a collision happening between any two vehicles moving on a two-dimensional surface. The index is the product of a longitudinal risk index $r_{lon} \in [0, 1]$ and a lateral risk index $r_{lat} \in [0, 1]$, which are piecewise linear functions that depend on the definitions of safe longitudinal and lateral distances, respectively. For the design of the risk indices, we use the safe distances presented in Shalev-Shwartz et al. (2017). The minimum safe longitudinal distance is

$$d_{min}^{lon} = \left[v_r \rho + \frac{1}{2} \rho^2 a_{max} + \frac{(v_r + \rho a_{max})^2}{2b_{min}} - \frac{v_f^2}{2b_{max}} \right]_+, \tag{5.1}$$

where $[x]_+ := \max\{x, 0\}$; c_r is a car with velocity v_r that drives behind another car c_f (in the same direction) with velocity v_f. For any braking of c_f of at most b_{max}, c_r has a response time of ρ during which it accelerates by at most a_{max}, and immediately after the response starts braking by at least b_{min} until necessary. Notice that the closer the two vehicles are to colliding, c_r will have to brake harder to avoid crashing, until its brakes reach their maximum capability (which we denote B) and a collision is unavoidable. When replacing b_{min} with B in d_{min}^{lon}, a tighter bound is defined for the safe longitudinal distance as

$$d_{min,brake}^{lon} = \left[v_r \rho + \frac{1}{2} \rho^2 a_{max} + \frac{(v_r + \rho a_{max})^2}{2B} - \frac{v_f^2}{2b_{max}} \right]_+. \tag{5.2}$$

Let d^{lon} be the current longitudinal distance between vehicles. We define the longitudinal risk

index

$$r_{lon} = \begin{cases} 0, & \text{if } d^{lon} \geq d^{lon}_{min} > 0 \\ 1 - \frac{d^{lon} - d^{lon}_{min,brake}}{d^{lon}_{min} - d^{lon}_{min,brake}}, & \text{if } d^{lon}_{min} \geq d^{lon} \geq d^{lon}_{min,brake} > 0 \\ 1, & \text{otherwise} \end{cases} . \tag{5.3}$$

Similarly, for the lateral analysis, the safe lateral distance is

$$d^{lat}_{min} = \left[\frac{v_{left} + v_{left,\rho}}{2}\rho + \frac{v^2_{left,\rho}}{2b^{lat}_{min}} - \left(\frac{v_{right} + v_{right,\rho}}{2}\rho - \frac{v^2_{right,\rho}}{2b^{lat}_{min}} \right) \right]_+ , \tag{5.4}$$

where, without loss of generality, car c_{left} is to the left of c_{right}, v_{left} and v_{right} are their respective lateral velocities, ρ is the same response time as in the longitudinal case, during which the two cars will apply a maximum lateral acceleration of a^{lat}_{max} toward each other, and after that both cars will apply lateral braking of at least b^{lat}_{min} until necessary; $v_{left,\rho} = v_{left} + \rho a^{lat}_{max}$ and $v_{right,\rho} = v_{right} - \rho a^{lat}_{max}$. As with the safe longitudinal distance, when replacing b^{lat}_{min} with B^{lat} (the maximum capable lateral braking of the vehicles) in d^{lat}_{min}, a tighter bound is defined for the safe lateral distance as

$$d^{lat}_{min,brake} = \left[\frac{v_{left} + v_{left,\rho}}{2}\rho + \frac{v^2_{left,\rho}}{2B^{lat}} - \left(\frac{v_{right} + v_{right,\rho}}{2}\rho - \frac{v^2_{right,\rho}}{2B^{lat}} \right) \right]_+ . \tag{5.5}$$

Let d^{lat} be the current lateral distance between cars. We define the lateral risk index

$$r_{lat} = \begin{cases} 0, & \text{if } d^{lat} \geq d^{lat}_{min} > 0 \\ 1 - \frac{d^{lat} - d^{lat}_{min,brake}}{d^{lat}_{min} - d^{lat}_{min,brake}}, & \text{if } d^{lat}_{min} \geq d^{lat} \geq d^{lat}_{min,brake} > 0 \\ 1, & \text{otherwise} \end{cases} . \tag{5.6}$$

Using minimum safe distances (MSDs) as a measure of safety risk assumes that maintaining these distances alone is sufficient to avoid collisions. However, estimating collision risk solely based on MSDs may overlook complex stochastic factors that influence collision probability. The literature presents alternative approaches, such as the Extreme Value Theory (EVT) (Ali et al., 2022) and the Generalised Pareto Distribution (GPD) (Zheng et al., 2014), to estimate collision risk by considering rare event statistics. While the MSD-based approach aligns with the objectives of this study, integrating EVT or GPD could yield further insights and is suggested for future research.

To create a unified risk index, the lateral and longitudinal risk indices are multiplied, with risk propensity parameters $\beta, \gamma > 0$ added to adjust sensitivity:

$$r = (r_{lon})^{\beta} (r_{lat})^{\gamma} , \tag{5.7}$$

which clearly lies in the range $[0, 1]$, and only takes a value > 0 when both longitudinal and lateral risk indices are also > 0 simultaneously. The risk propensity parameters β and γ adjust the weighting of the longitudinal and lateral risk indices within the overall risk index. Values of $\beta, \gamma > 1$ reduce the risk index, representing a lower sensitivity to minor safety distance violations and reflecting a more aggressive driving style. Conversely, values $\beta, \gamma < 1$ increase the risk index, making the controller more sensitive to safety distance reductions, thereby representing a risk-averse behaviour. This flexibility allows the risk model to adapt to different

driving behaviours or operational requirements.

Proposition 1. *$r = 0$ if and only if a collision is impossible to occur in the future given the current states of the vehicles (i.e. positions, velocities and accelerations) and following the braking and response time assumptions.*

Proof. $r = 0 \iff r_{lon} = 0 \vee r_{lat} = 0 \iff (d^{lon} \geq d^{lon}_{min} > 0) \vee (d^{lat} \geq d^{lat}_{min} > 0)$, meaning at least one safety distance is being held. □

Proposition 2. *If $r = 1$, avoiding a potential collision cannot be guaranteed under the assumptions of safe longitudinal and lateral distances, independent of the risk propensity parameters β and γ.*

Proof. $r = 1 \iff r_{lon} = 1 \wedge r_{lat} = 1 \iff (\geq d^{lon} \geq d^{lon}_{min,brake}) \wedge (\geq d^{lat} \geq d^{lat}_{min,brake})$, meaning both the longitudinal and lateral safety distances under maximum braking are being violated. If the other vehicle brakes, a collision can not be guaranteed to be avoided under the model conditions. □

5.3 Learning approach

This section presents our proposed model for the safe and efficient passing of EMVs in autonomous traffic. A diagram can be found in Figure 5.2. Most studies have addressed this problem based on rules and heavy simplifications of vehicle-to-vehicle interactions. Rule-based methods cannot cover all possible situations in autonomous traffic; therefore, we adopt a Multi-Agent Reinforcement Learning approach. MARL allows us to simulate and train agents on thousands of different scenarios and achieve effective coordination among agents. In our formulation, all AVs and the EMV are represented as learnable agents that can take independent actions based on partial observability of the environment. We adopt the following assumptions and simplifications:

Assumption 1. ***The environment considers a set of high-level discrete actions.*** *In this study, we are interested in the multi-agent coordination policies, rather than the ability of*

individual vehicles to follow the road and stay on track. Thus, we use a discrete action space that considers high-level actions; accelerate, brake and change lane. This is a common assumption of multi-agent autonomous driving simulators (Zhou et al., 2020a; Palanisamy, 2019a; Zhou et al., 2022), where vehicle dynamics are considered part of the environment.

Assumption 2. ***The autonomous vehicles that have been considered in this study are assumed to possess SAE Level 4 autonomy or above.*** *SAE Level 4 autonomy (SAE, 2019) represents a highly automated driving system, enabling the vehicle to operate independently and handle all driving tasks within a predefined operational design domain (ODD) without the need for human intervention. At this level of autonomy, the vehicle demonstrates the ability to navigate most driving situations and scenarios autonomously, although it may have specific limitations or constraints in certain environments or under particular conditions.*

Assumption 3. ***All agents have partial observability of the environment, assuming accurate and consistent representation in their observations.*** *Fixed-range AV sensors and the stochastic nature of vehicle-to-vehicle interactions cause AVs to have a limited perception of their surrounding environment (Lecerf et al., 2022). We adopt a fixed range of 70m, which is realistic according to current communication protocols (Kenney, 2011). It is important to acknowledge that one assumption in this study is that the partial observations provided are always correct, as accurate information is crucial for MARL-based EMV optimisation for ensuring safe and effective decision-making. The incorrect or inaccurate sensor data is a more general challenge for the AV sector and is beyond the immediate scope of this study. Having established this model, our future efforts will concentrate on incorporating fault-tolerant mechanisms, such as redundant sensors or sensor fusion techniques, to alleviate the effects of sensor failures in the high-speed scenarios that would be encountered by EMVs. Additionally, the exploration of robust RL algorithms capable of effectively handling missing or noisy state information is of utmost importance.*

Assumption 4. ***EMVs and AVs are controlled with the same type of policy but different reward structure.*** *Given the focus of this study on strategic manoeuvring, we use the same type of policies for controlling EMVs and AVs. Namely, policies with the same type of*

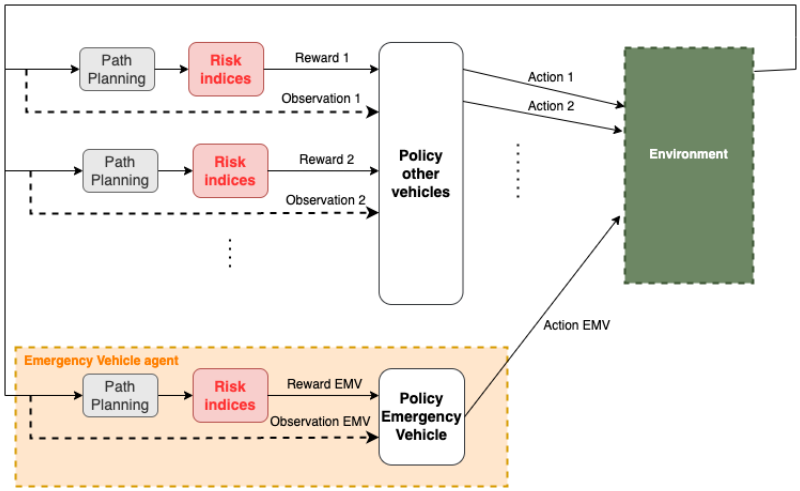


Figure 5.2: Diagram of the risk-aware learning methodology

observation inputs and action outputs and with the same neural network architecture. However, their policies are trained with different rewards to allow for the safe passage of the EMV.

5.3.1 Multi-Agent Deep Reinforcement Learning

We formalise the MARL problem as a Partially Observable Stochastic Game (POSG). As outlined in Chapter 2, this is defined by the tuple $\langle N, S, \mathcal{A}, T, \mathcal{R}, O, \gamma \rangle$, where N is the number of agents, O is the set of individual observations for each agent, S is the set of states, $\mathcal{A}$ is the set of joint actions, T is the transition probabilities, $\mathcal{R}$ is the reward function and γ is the discount factor.

We implement the Multi-Agent Policy Optimisation algorithm (MAPPO), a state-of-the-art actor-critic MARL method that has shown to achieve superior performance in several coordination-heavy tasks (Yu et al., 2021). Proximal Policy Optimisation is a gradient-based method that approximates the policy directly. It includes a clipping objective L^{Clip} that controls the policy update during the training period. This forces the policy update to be conservative if it deviates too much from the current policy. The actor and critic update equations for MAPPO are presented in the Appendix.

In contrast to the single-agent version, in MAPPO, all agents share a centralised critic during the training phase, but only the actors are needed in execution. Thus, it belongs to the Centralised

Training Decentralized Execution (CTDE) class of algorithms. MAPPO is an on-policy method. Therefore, it does not rely on previous experience. This can make it less sample efficient than off-policy methods such as MADDPG (Lowe et al., 2020) and QMIX (Rashid et al., 2018a), at least in theory. In practice, MAPPO has been found to be a powerful and robust method in several cooperative environments, even outperforming state-of-the-art off-policy algorithms (Yu et al., 2021). The implemented MAPPO algorithm is presented in Algorithm 1.

Algorithm 1 MAPPO for autonomous traffic control

Initialize θ, the parameters for the critic V and the parameters ϕ for the actor π.
for $ep = 1$ to M **do**
  Retrieve initial state s_0, O_0: Individual speeds, positions, lanes and perception of surrounding vehicles.
  for $t = 1$ to T **do**
    for j in *agents* **do**
      Choose high-level action $a_j^t = \pi(O_j^t; \theta)$
      Calculate value $v_j = V(s_j^t, \phi)$
    end for
    Execute high-level actions and calculate new pose based on vehicle dynamics.
    Observe r^t, s^{t+1}, O^{t+1}.
    Store $[s^t, O^t, a^t, r^t, s^{t+1}, O^{t+1}]$.
    Compute policy loss: $L^{\text{CLIP}}(\theta) = E^t\left[\min\left(r^t(\theta)\hat{A}^t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^t\right)\right]$,
    where $r^t(\theta) = \frac{\pi_\theta(a^t|s^t)}{\pi_{\theta_{\text{old}}}(a^t|s^t)}$
    Compute value loss: $L^{\text{V}} = E^t\left[(V(s^t;\phi) - \hat{R}^t)^2\right]$
    Compute advantage estimate $\hat{A}$ via GAE using the stored transitions.
    Compute reward-to-go $\hat{R}$ on the stored transitions.
  end for
  Update θ using Adam optimiser on mini-batch
  Update ϕ using Adam optimiser on mini-batch
end for

A graphic representation of the MAPPO approach for autonomous traffic is presented in figure 5.3. In the first instance, we consider a single critic for all agents during the training phase. A separate actor network allows agents to take individual actions during execution, without the need for a centralised state. Therefore the implementation of the proposed method results in decentralised multi-agent policies. We also explore using a separate critic network for each agent in a competitive version of the environment.

We represent the EMV as an actor taking independent actions in the environment, although we acknowledge it can be an autonomous or a human-driven vehicle. It should be noted that given

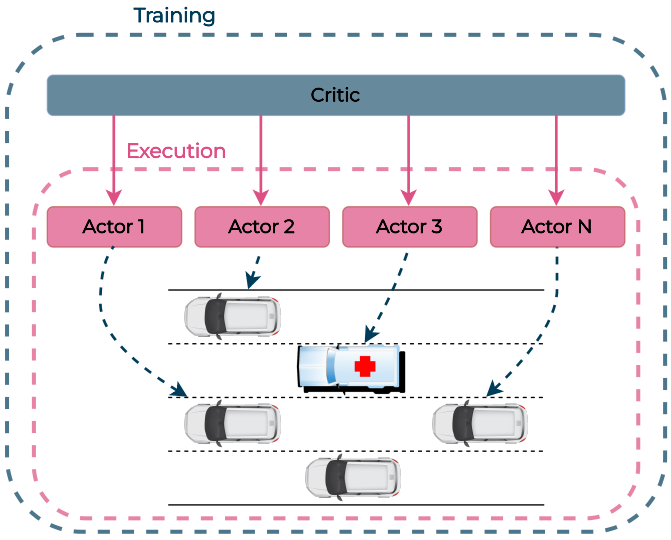


Figure 5.3: Graphic representation of the multi-agent autonomous traffic approach. Each AV as well as the EMV correspond to an actor network. During the training phase, all agents share a centralised critic. During execution, the critic is no longer needed.

the challenging and critical nature of emergency driving, it is not expected to have reliable autonomous EMVs in the coming years. Nevertheless, improving the safety of AV systems in the presence of EMVs is a crucial first step in the reliable deployment of AVs.

5.3.2 Emergency Vehicle Multi-Agent Environment

To train and test Reinforcement Learning (RL) methods, the OpenAI Gym framework (Brockman et al., 2016) is the standard approach. It offers a simulated environment where agents learn policies through continuous interaction. Our work builds upon the multi-agent autonomous driving environment developed by Candela et al. (2022) by introducing EMVs and risk metrics. We also expanded the agents' action space to include heavy braking and accelerations.

We present the Emergency Vehicle Multi-agent Autonomous-driving (EMMA) environment, which enables the testing of Reinforcement Learning policies in various driving scenarios. Figure 5.4 shows a representation of the environment. In contrast to other gym environments for Autonomous Traffic, EMMA considers all vehicles to be trainable agents with partial observability of their surroundings. Each vehicle has state defined by a 6D tuple of its 2D position (x, y), orientation (θ), velocity $(\dot{x}, \dot{y})$ and angular rate $(\dot{\theta})$. Each agent has kinematic constraints

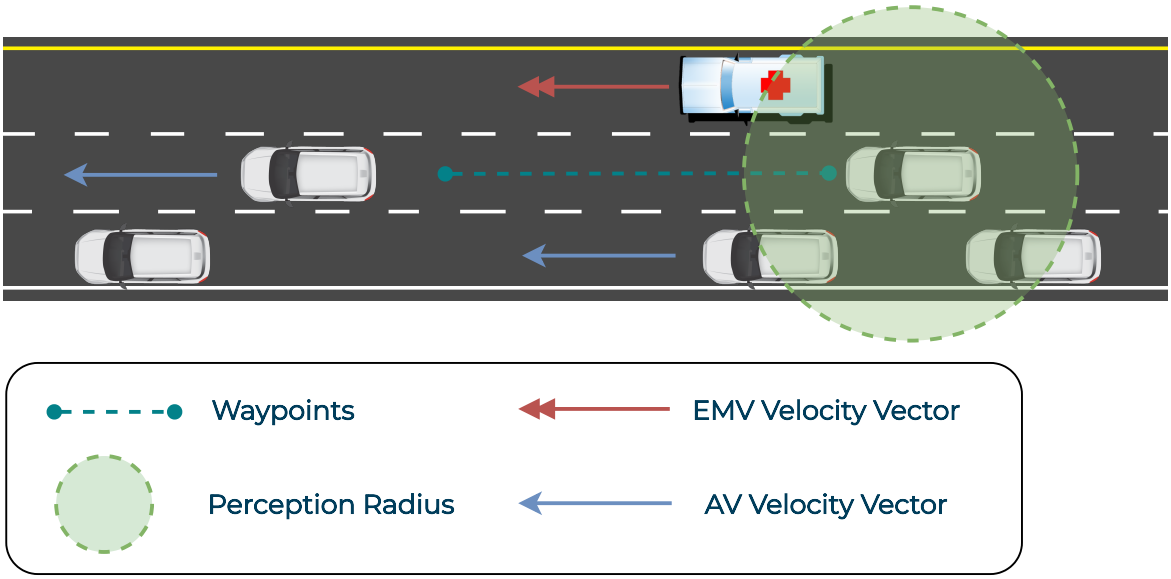


Figure 5.4: Representation of the multi-agent environment. The road layout (waypoints) is defined by the user, and the perception radius follows Assumption 2.

(e.g. max braking a_{max}) implemented as parameters based on the vehicle's type. At each simulation step, agents receive individual local observations from the environments. These are the inputs to the MARL policy, which outputs an independent optimal action for each vehicle. A nonlinear kinematics model (Ercan, 2019) then updates the vehicles' state based on the selected action. Finally, each agent receives an individual reward as defined in equation 5.8. A waypoint-based system is implemented on each track to enable lane changing behaviour and to facilitate the calculation of metrics with respect to cars on the same lane and on adjacent lanes.

To ensure the safe and efficient transit of EMVs, the reward function incorporates the risk metrics mentioned in the previous subsection and performance metrics related to the speed of each vehicle. It further includes the speed and risk metrics of the EMV to incentivize cooperation between agents in order to prioritize the EMV. Compared to regular AVs, EMVs have different dimensions and a greater speed limit. Additional details behind the logic and architecture of the EMMA environment are presented in Appendix A.

We define the individual elements of the environment as follows:

- Agents N: There are two types of agents: AVs and the EMV. The AVs have the same

physical characteristics, range of speeds and perception range. The EMV has different physical characteristics and is allowed to go at higher speeds than the AVs.

- Observation O: The local observation of the AVs consists of their own speed, position, lane, and a limited radial perception of surrounding vehicles (determined by Assumption 2). Thus, AVs can recognise when an EMV is approaching at a certain distance but cannot see its global position.

- Action A: All agents have a discrete action space with seven possible high-level decisions—accelerate; brake; heavy acceleration; heavy brake; change lane to the left; change lane to the right; keep previous velocity (no acceleration/braking). If an agent tries to perform a lane change but cannot for physical reasons, the action does nothing, and agents maintain their lane and speed.

- Reward $\mathcal{R}$: The reward function for vehicle i is defined by the following equation:

$$\mathcal{R}_i = r_i + v_i + p_{i,col} + p_{i,lcm} + c(v^{EV} + p^{EV}_{lcm}) \tag{5.8}$$

The reward function in equation (5.8) is constructed as follows: r_i is the ith agent's risk of collision as defined by equation (5.7), v_i is its current speed measured in m/s, $p_{i,col}$ is a collision penalty and $p_{i,lcm}$ is a penalty for lane changing manoeuvres. Lane changes can cause a frontal collision for the following vehicle (Fitch et al., 2009), and sudden lane changes increase the likelihood of collisions 2.53 times (Shawky, 2020). Therefore, we aim to discourage lane changes. However, the agent will learn to change lanes if the expected gain in speed is larger than the penalty for changing lanes.

In addition, v^{EV} is EMV's current speed and p^{EV}_{lcm} corresponds to an additional penalty for lane changing manoeuvres conducted by the EMV. These last two components of the reward function guarantee cooperation among agents in the presence of the EMV. The collaboration level is adjusted by collaboration factor $c \in [0, 1]$ — if $c = 0$ the agents are not incentivised to aid the EMV.

We normalised the individual components of the reward function between 0 and 1, except for the collision penalty, which we set to a very low value (-100) because of their relatively low frequency.

Our environment EMMA is sufficient for validating our strategic decision-making framework since low-level vehicle control is out of the scope of this study. The non-linear dynamics model by Ercan (2019) and the environment's road layout description are accurate enough for strategic and decision-making purposes. In addition, it provides different road representations that resemble challenging real-world scenarios.

Nevertheless, EMMA has some limitations on the traffic conditions it can simulate, which we leave as future work: 1) all cars have the same observability radius and receive perfect state information. In contrast, different sensor suites and higher-fidelity communication between cars of the same manufacturer could lead to different sensing capabilities; 2) all regular AVs are taken to have the same kinematic parameters and geometry, while cars from different automakers or types would behave slightly differently; 3) weather and road conditions are not currently considered, and these lead to different effective kinematic parameters (e.g. wet roads lead to a decrease in the maximum safe deceleration); 4) agents cannot communicate with other vehicles to share their intent or observations; 5) any vulnerable road users (VRU) such as bicycles and pedestrians; 6) there can only be up to one EMV in a given scenario.

5.4 Analysis

The proposed MARL method was tested on a long track, where agents can go in all lanes in a single direction. We test two different scenarios: a two-lane road and a four-lane highway. However, the environment can be easily adapted to a multiple-lane setting with traffic going in both directions. We leave the study of other road layouts, such as intersections and roundabouts, as future work. The EMV has to make its way through the traffic by either overtaking or by AVs moving to another lane. First, we evaluate the performance of the proposed method, comparing it against two car following models. Second, we consider different variations of the

reward function and evaluate the trade-off between collision risk and traffic efficiency. Third, we assess the scalability of the proposed method by evaluating scenarios with a different number of agents. Finally, we present a fully competitive case where AVs have their own independent reward and neural networks. Implementation details and simulation parameters are presented in Appendix A.

5.4.1 Comparison to baselines

We compare the proposed MARL approach against three baselines: the Gipps car-following model (Gipps, 1986), which we use to simulate human driving, an Adaptive Cruise Control (ACC) method based on Model Predictive Control (MPC) (Zhu et al., 2020b) and the Deep Q-Network (DQN) MARL method. We chose to use the driver behaviour models instead of traffic micro-simulation tools such as SUMO or VISSIM because of the increased transparency and customisation achieved by directly implementing the models. In commercial micro-simulation tools, gaining full control of the vehicle's motions and manoeuvres remains unfeasible or impractical. Given the nature of this study, we deemed the latter to be an essential requirement of the modelling approach that we had to follow and, therefore, had to implement the driving models ourselves. Furthermore, the traffic models of most simulation packages are comprised of adaptations of Gipps' model or other similar ones (Spyropoulou, 2007). For the DQN baseline, we used the parameters presented by Dong et al. (2020), which demonstrated strong performance in environments with human-driven vehicles. Details on the DQN architecture can be found in Appendix B.

Gipps Car-Following Model

The Gipps model is a well-known car-following approach that corresponds to the safe-distance class of models. It is used in many leading microsimulation tools as a default human driving behaviour due to its simple calibration (Spyropoulou, 2007; Barceló and Casas, 2005). We chose this model because it considers the risk of potential collision between vehicles and is simple to

calibrate (Ahmed et al., 2021). In this model, the actions to change lanes are governed by a fixed set of rules. The rules consider the risk of collision, the presence of vehicles in the other lane and the potential gain in the average speed of the decision. The main disadvantage of this model is that it does not consider vehicle perception (Lazar et al., 2016). In addition, minor variations on the leader's vehicle can heavily affect the followers' reaction.

Gipps model determines the speed of each vehicle at time t by applying two constraints on the acceleration: the desired speed must not be surpassed, and the critical safe distance should be maintained. The acceleration and deceleration can be expressed by the following equations:

$$v_n^{acc}(t+T) = v_n(t) + 2.5 \cdot a_n \cdot T(1 - \frac{v_n(t)}{v_n^d}) \cdot \sqrt{0.025 + \frac{v_n(t)}{v_n^d}} \tag{5.9}$$

$$v_n^{decc}(t+T) = -T \cdot d_n + \sqrt{T^2 \cdot d_n^2 + d_n\{2x_{n-1}(t) - x_n(t) - (S_{n-1}) - T \cdot v_n(t) + \frac{v_{n-1}(t)^2}{d'_{n-1}}} \tag{5.10}$$

where n and $n-1$ are the follower and leader, respectively, T is the reaction time, $v_n(t)$ and $v_{n-1}(t)$ are the speeds of the follower and leader respectively at time t. v_n^d and a_n are the follower's maximum desired speed and maximum acceleration, respectively. d_n is the most severe braking capability of the follower, and d'_{n-1} is the follower's estimate of the leader's breaking rate. $x_n(t)$ and $x_{n-1}(t)$ are the longitudinal positions of the follower and leader at time t. S_{n-1} is the distance between vehicles at a stop.

The implemented version of the Gipps model uses safe following distances set according to the RSS model introduced by Shalev-Shwartz et al. (2017). The reaction time was set to $T = 0.7s$. To ensure realistic traffic flow simulation with the Gipps model, we based our parameter selection on the already calibrated simulated parameters established by Punzo and Tripodi (2007). Their study developed a calibrated Gipps model using real motorway data and identified a combination of parameter values that delivered consistently successful results

against the traffic flow data. We assume that all human-driven vehicles have the same behaviour, represented as a light vehicle driving through a fast lane with the following parameters $T = 0.2, v_n^d = 80km/h, d_n = 1.5$ and $d'_{n-1} = 2.0$

Adaptive Cruise Control based on MPC

This subsection describes the ACC method based on MPC, which is used for autonomous vehicle control (Zhu et al., 2020b). The method was adapted from the one developed by Zhu et al. (2020b) for a multi-agent setting and discrete action space. The model considers a kinematic point mass model that can be described by the following equation:

$$x(t+1) = Ax(t) + Bu(t) \tag{5.11}$$

where $x(t) = (S_{n-1,n}(t), \Delta V_{n-1,n}(t), V_n(t))^T$, $u(t) = a(t)$ and

$$A = \begin{bmatrix} 1 & \Delta T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.12}$$

$$B = \begin{bmatrix} -0.5\Delta T^2 \\ -\Delta T^2 \\ \Delta \end{bmatrix} \tag{5.13}$$

The objective of an ACC is to follow the leader vehicle (LV) at the desired distance $\tilde{S}_{n-1,n}$, considering the vehicle speed and the headway t_{hw}:

$$\tilde{S}_{n-1,n} = V_n t_{hw} \tag{5.14}$$

The mathematical formulation of the MPC model is presented below:

$$\min_{a} \quad \sum_{t=0}^{N-1} [(\frac{S_{n-1,n}(t) - \tilde{S}_{n-1,n}(t)}{S_{max}})^2 + (\frac{\Delta V_{n-1,n}}{\Delta V_{max}}) + (\frac{jerk(t)}{jerk_{max}})^2] \tag{5.15a}$$

$$\text{s.t.} \quad x(t+1) = Ax(t) + Bu(t), \tag{5.15b}$$

$$S_{n-1,n} > 0, \tag{5.15c}$$

$$V_n > 0, \tag{5.15d}$$

$$-3m/s^2 \leq a_n \leq 3m/s^2 \tag{5.15e}$$

where N is the prediction horizon, S_{max}, ΔV_{max} and $jerk_{max}$ are constants. Constraint (5.15b) corresponds to the point-mass model equations. Constraint (5.15c) ensures that the relative distance between the leader and the following vehicle is always greater than 0. Constraints (5.15d) and (5.15e) state that the speed of the following can't be negative, and its acceleration should be within a predefined range, respectively. We adopt the same parameter values used by Zhu et al. (2020b), that is, $t_{hw} = 1.2s$, a $S_{max} = 15m$, $\Delta V_{max} = 8m/s$. In addition, we set the jerk term to be equal to zero because the MARL formulation does not consider it as part of the objective.

Results Comparison

Table 5.1 presents the comparison results between the proposed MARL method, the Gipps lane changing model, the MPC method and the DQN method. We used a fixed number of AVs for this set of experiments and trained the algorithm for 2000 episodes. The results shown in the table correspond to averages over 10 test episodes. The proposed method consistently

overperforms both the Gipps and MPC methods. It produces policies with higher average speeds for the EMV while maintaining safer distances between all vehicles. The collision risk determined by the risk index is also considerably lower for the MARL method. The average number of lane changing manoeuvres (LCM) for the Gipps model is considerably higher when compared to the proposed model; this is because the Gipps model doesn't penalise the lane changes of the EMV.

Table 5.1: Comparison to the car following baselines.

Scenario	Method	Reward	EMV Speed	AV Speed	Col. rate	Col. risk	Safety Dist. (m)	Avg. EMV LCM
Road	Gipps	-452.1 ± 280.0	19.8 ± 0.4	$\mathbf{19.6 \pm 0.3}$	15.3 ± 10.2	0.31 ± 0.05	6.5 ± 0.9	87.0
	MPC	401 ± 17.1	14.3 ± 0.5	14.1 ± 0.2	1.0 ± 3.0	$\mathbf{0.2 \pm 0.4}$	16.7 ± 4.0	38.6
	DQN	328 ± 62.3	18.3 ± 0.9	13.9 ± 1.1	3.1 ± 1.5	0.28 ± 0.06	17.2 ± 3.8	34.5
	MAPPO	$\mathbf{653 \pm 67.1}$	$\mathbf{22.7 \pm 1.4}$	19.1 ± 0.4	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.01	$\mathbf{17.7 \pm 3.6}$	32.1
Highway	Gipps	229.2 ± 110.2	19.58 ± 0.4	19.4 ± 0.2	5.1 ± 2.0	0.3 ± 0.06	17.0 ± 1.1	96.3
	MPC	523.4 ± 96.1	16.3 ± 0.3	14.9 ± 0.2	0.0 ± 0.0	0.1 ± 0.02	19.3 ± 3.6	48.7
	DQN	475.1 ± 70.2	17.5 ± 0.4	14.5 ± 0.3	0.1 ± 0.1	0.12 ± 0.03	16.4 ± 2.9	46.2
	MAPPO	$\mathbf{877.2 \pm 53.4}$	$\mathbf{29.9 \pm 0.9}$	$\mathbf{19.8 \pm 0.3}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.03 \pm 0.002}$	$\mathbf{37.0 \pm 7.0}$	53.3

For the DQN method, results show moderate performance compared to MPC and MAPPO. While the EMV speed in the DQN method is slightly higher than that of MPC, the overall performance in terms of collision risk and safety distance is lower. DQN also exhibits higher collision rates and a marginally higher risk index compared to MPC, indicating that although DQN allows faster transit for the EMV, it does so with slightly compromised safety.

In contrast to the car following models, the proposed method allows for the efficient transit of the EMV in heavy traffic conditions. Thus, the EMV can achieve higher speeds while keeping safe distances from other vehicles. Figure 5.5 shows the distribution of average speeds for the EMV and the AVs for the three methods. The speed of the AVs is slightly lower than the maximum allowable speed ($20m/s$) because AVs can effectively collaborate and slow down to facilitate the passage of the EMV. On the other hand, the Gipps model does not prioritise the passage of the EMV because it maximises the speeds of all vehicles. The MPC method achieves nearly no collisions with a low overall risk of collision; however, the average speeds of the vehicles are also considerably lower.

The proposed method guarantees the safe transit of the EMV by incorporating the risk metric

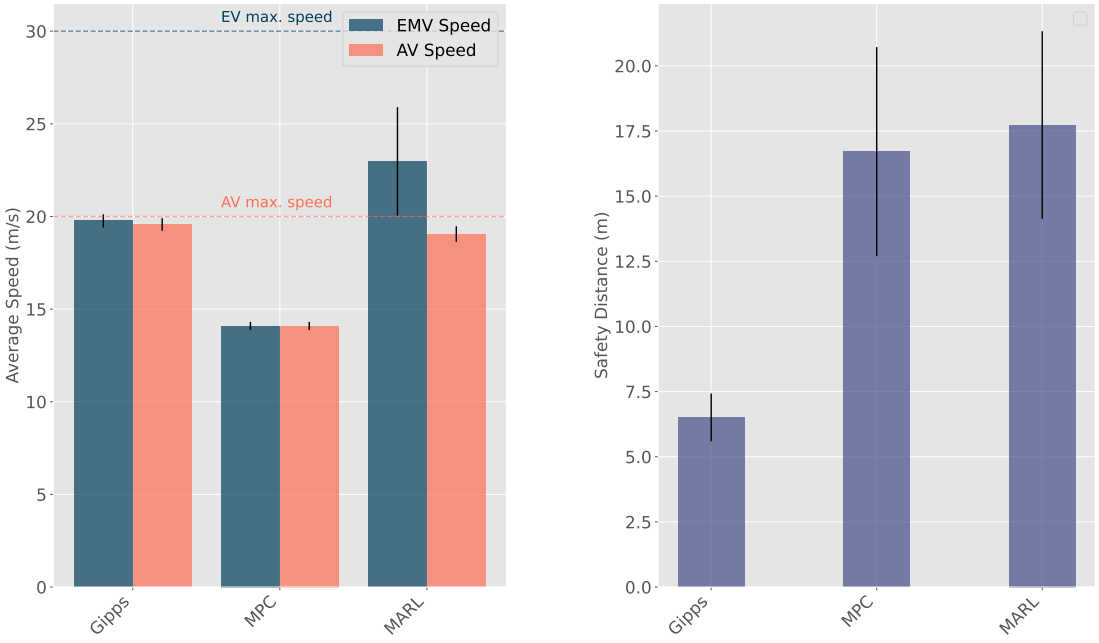


Figure 5.5: Comparison of the policies obtained with the three different methods. (a) Average speeds for the EMV and AVs and (b) Safety distances

in the reward. No collisions occurred during the testing episodes, and the average collision risk is substantially lower than the Gipps method for both scenarios. The proposed method achieves a low collision risk thanks to preemptive manoeuvring actions from the AVs when the EMV is within their perception range. An example of these types of actions is shown in figure 5.6.

The MPC method achieves a low number of collisions and the lowest overall risk of collision. However, it is conservative in terms of speed, which causes it to perform worse in terms of total reward when compared to the MARL method.

Figure 5.7 displays the speed profiles of AVs and the Emergency EMV under two control methods: MPC (left) and MAPPO (right). In the MPC profile, both the AVs and EMV maintain conservative speeds of around 14 m/s after the EMV's arrival, reflecting MPC's focus on safety and stability. The minimal variation between the speeds shows a cautious approach that avoids aggressive driving, ensuring safe but slower manoeuvres.

In contrast, the MAPPO profile allows for faster EMV speeds (22-23 m/s) while AVs maintain around 19 m/s. MAPPO enables higher speeds while preserving safe distances between vehicles,

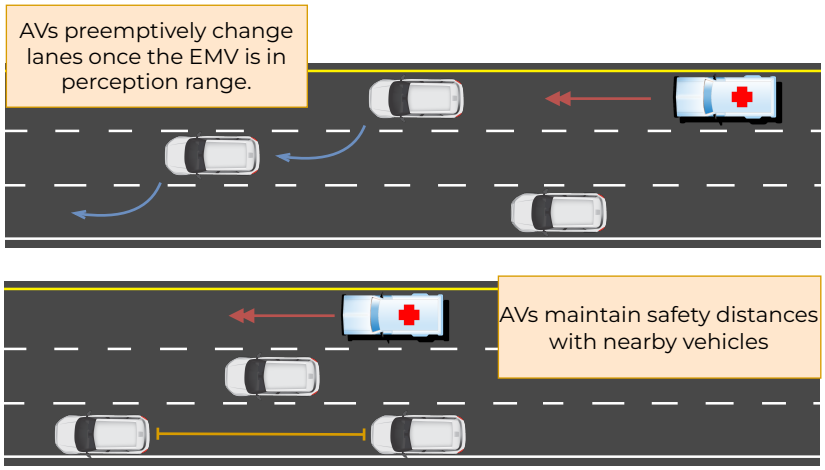


Figure 5.6: Example of preemptive lane changing maneuvers by the AVs.

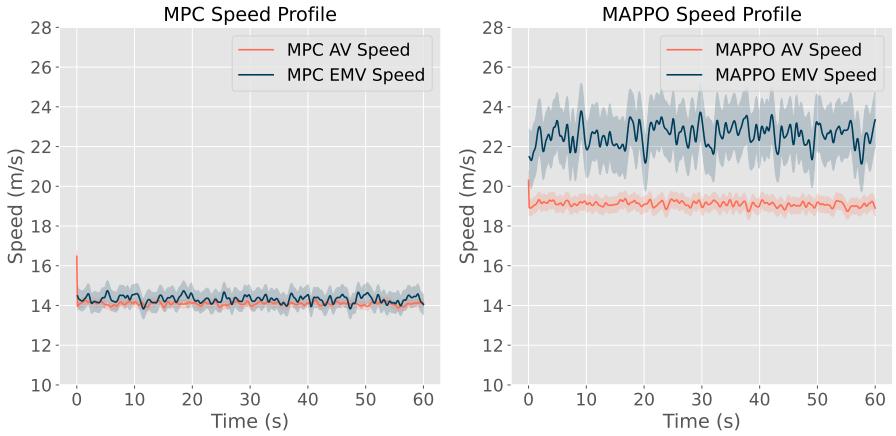


Figure 5.7: Speed Profiles for MPC and MAPPO methods. The speed profiles show the average speed of the Autonomous Vehicles and the EMV under two control methods: Model Predictive Control (MPC) and Multi-Agent Proximal Policy Optimization (MAPPO)

highlighting its efficiency in balancing speed and safety compared to the more conservative MPC approach.

5.4.2 Reward Design: Risk vs Efficiency

We performed different experiments by varying the two main components of the reward function: risk of collision and traffic flow efficiency. The risk component is determined by equation 5.7, meanwhile the efficiency is determined by the average speed. If two vehicles are within the range of collision risk as defined by equation 5.6, then an increase in speed has a negative

impact on the risk index. Conversely, if the speed decreases, the risk index improves. Thus, there is a trade-off between the speed and the risk of collision.

Table 5.2 presents the results for different combinations of risk and efficiency within the reward function. We used a fixed number of agents (30) for these experiments and trained the MAPPO algorithm for 2000 episodes. The results shown in table 5.2 correspond to averages over 10 test episodes. As expected, the speeds of both the EMV and the AVs are very high when the risk component is reduced. However, the collision rate and the collision risk are considerably worse compared to risk-aware policies (when the risk coefficient is greater than 0). In addition, the average safety distance between vehicles is lower when risk is not present in the reward function.

Table 5.2: Results of different reward functions.

Risk	Efficiency	EV Speed (m/s)	AV Speed (m/s)	Col. rate	Col. risk	Safety Dist. (m)
0.0	1.0	27.83 ± 1.16	19.77 ± 0.24	3.90 ± 1.40	0.189 ± 0.013	16.95 ± 3.82
0.5	1.0	18.86 ± 0.81	17.86 ± 0.16	4.01 ± 3.21	0.176 ± 0.021	17.03 ± 2.87
1.0	1.0	7.22 ± 0.27	7.22 ± 0.11	0.8 ± 2.4	0.114 ± 0.031	16.69 ± 3.76
1.0	0.5	7.14 ± 0.18	7.12 ± 0.16	0.0 ± 0.0	0.105 ± 0.015	16.23 ± 4.61
1.0	0.0	7.13 ± 0.33	7.09 ± 0.055	0.0 ± 0.0	0.094 ± 0.019	16.36 ± 3.64

Figure 5.8 shows the effect of increasing the risk component in the reward function over the EMV average speed and the risk of collision. The speed of the EMV is reduced drastically as the risk component approaches the value of 1.0. Moreover, when the risk component reaches the value 1.0, the average speed is equal to the minimum velocity ($7m/s$). This is because the speed has a direct negative impact over the safe lateral and longitudinal distances, according to the risk model equations.

5.4.3 Mixed traffic

Since there is no definitive timeline for the widespread adoption of AVs (Palade and Deo, 2022), it is imperative to develop collaborative control methods that can facilitate the coexistence of both autonomous and human-driven vehicles (HDV). To achieve this objective, we executed a series of experiments that involved a combination of AVs and human-driven vehicles (HDVs). In each experiment, we trained the AVs with the proposed MARL pipeline, while the behaviour

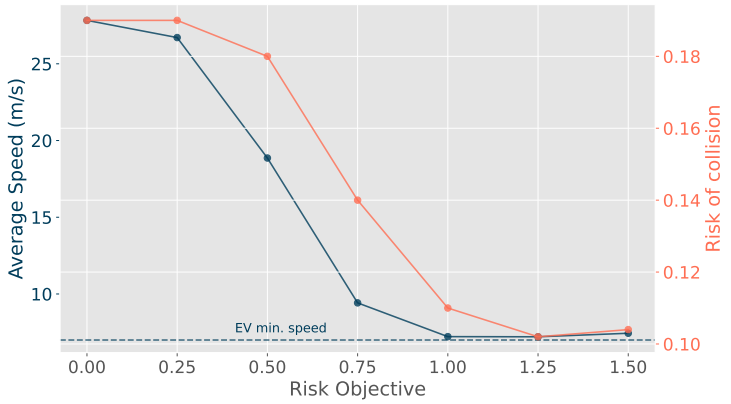


Figure 5.8: Influence of the risk component over the EMV speed and the risk of collision. The risk component is determined by equation 5.7.

of the HDVs was simulated using the Gipps method. Initially, we started with a small number of HDVs and gradually increased their proportion in the fleet until they constituted 80% of the total number of vehicles. The number of AVs and HDVs used in the experiments ranged from 2 to 8, with different ratios of AVs to HDVs.

The results of these experiments are summarized in table 5.3, in which it can be observed that a larger proportion of AVs results in better performance in general: higher EMV speed, lower collision rate and lower collision risk across all vehicles, and fewer lane changes for the EMV.

Table 5.3: Results of the experiments for different mixed scenarios.

N° AVs	N° HDVs	EMV Speed (m/s)	AV Speed (m/s)	Col. rate	Col. risk	Safety Dist. (m)	Avg. EMV LCM
8	2	24.86 ± 3.12	19.45 ± 2.35	5.35 ± 2.41	0.176 ± 0.071	15.12 ± 5.47	35.12
6	4	19.88 ± 4.35	19.92 ± 3.31	4.76 ± 3.22	0.189 ± 0.067	16.49 ± 4.29	40.24
4	6	18.78 ± 3.67	19.53 ± 4.47	7.14 ± 2.21	0.210 ± 0.098	14.95 ± 3.83	72.12
2	8	18.56 ± 3.68	19.23 ± 5.77	9.56 ± 4.35	0.267 ± 0.112	14.45 ± 4.61	91.15

For instance, in the first experiment, where there were 8 AVs and 2 HDVs, the average collision rate was 5.35. As the number of AVs decreased, the collision rate increased, with the highest collision rate of 9.56 occurring in the last experiment, where there were only 2 AVs and 8 HDVs. This suggests that the proposed methods can potentially improve safety in mixed traffic scenarios by reducing the number of collisions. The safety improvement will be larger while the ratio between AVs and HDVs increases.

The number of lane changing manoeuvres (LCM) increases when more HDVs are introduced;

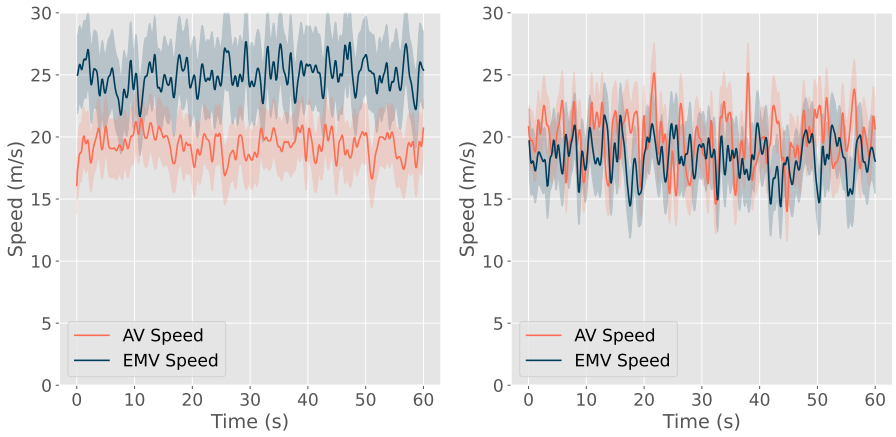


Figure 5.9: Speed Profiles for MAPPO-Controlled Autonomous Vehicles (AVs) with Human-Driven Vehicles (HDVs). The left graph shows a scenario with 8 AVs and 2 HDVs, while the right graph represents a scenario with 2 AVs and HDVs. In both cases, the Emergency Vehicle (EMV) speed is depicted alongside the AV speed.

this is because the Gipps model doesn't penalise the lane changes of the EMV. This can be associated also with the higher collision rate, as suggested by Shawky (2020).

Figure 5.9 illustrates two scenarios where MAPPO controls Autonomous Vehicles (AVs), and Human-Driven Vehicles (HDVs) are simulated using a car-following model. On the left, the scenario involves 8 AVs and 2 HDVs. Here, the MAPPO-controlled AVs maintain relatively high and stable speeds (around 20 m/s), while the EMV operates at an even higher speed (around 25 m/s). This scenario demonstrates how MAPPO efficiently manages a larger number of AVs while enabling the EMV to maintain high speeds and safe interactions with other vehicles.

On the right, the scenario includes only 2 AVs alongside 8 HDVs. In this case, the MAPPO algorithm is more conservative, as the AVs exhibit more variability in their speeds and travel at lower speeds to accommodate manoeuvres around the HDVs and the EMV. Despite this, the EMV still achieves relatively high speeds, indicating that the algorithm prioritises its rapid transit while adjusting AV behaviour to ensure safe navigation in a more constrained environment. This contrast highlights the algorithm's adaptability in balancing speed and safety across varying levels of vehicle interaction.

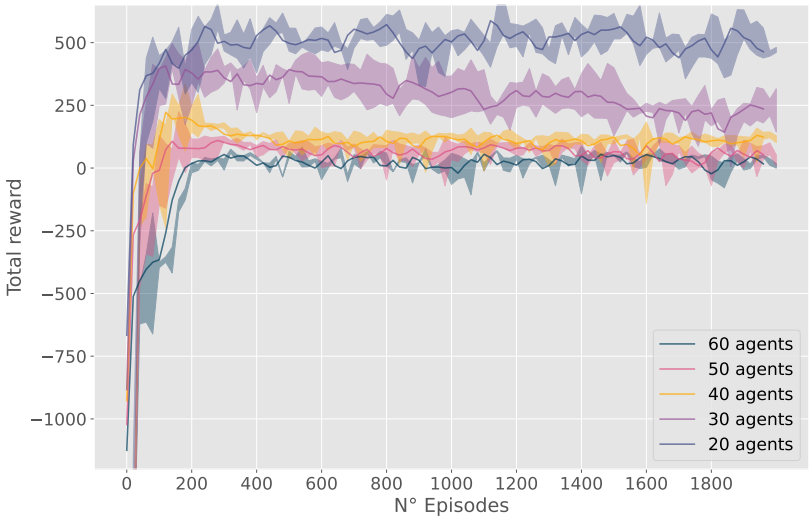


Figure 5.10: Learning curves for different number of agents.

5.4.4 Scalability

To test the scalability of the proposed method, we explored the effect of using a different number of agents over the training process and the performance of the trained policies. Additional agents do not alter the state dimension because agents have partial observability of the environment. Thus, a larger number of agents does not translate into higher neural network complexity. However, a larger number of agents produces higher environment complexity and, therefore, longer training time.

Figure 5.10 shows the effect of different numbers of agents over the learning curves. The reward corresponds to the sum of rewards for all agents. For each case, four seeds were used, and the Figure shows the average and interval. Due to the higher number of vehicles on the road, the total average reward converges to considerably lower values when increasing the number of agents.

Table 5.4 shows the individual results for the trained multi-agent policies with a different number of agents. Results correspond to the average over 10 test episodes. All vehicles' initial positions and speeds were randomised for every test episode. When increasing the number of agents, AVs and the EMV must move at lower speeds, which causes the individual rewards to

be lower. Moreover, vehicles will be closer to each other, which reduces the safety distance and thereby increases the risk of collision. The last column of the table contains the training time for the 2000 episodes, expressed as Frames Per Second (FPS). The total training time can be obtained by dividing the total number of steps by the FPS. For example, for the case with 20 agents, the total training time is $t = (2000 \cdot 400)/(31 \cdot 3600) = 7.17$ hr.

Table 5.4: Results for different number of agents.

N° agents	Reward	EMV Speed	AV Speed	Average Dist. (m)	Col. risk	FPS
20	563.29 ± 30.00	26.89 ± 0.49	18.68 ± 0.15	17.03 ± 3.87	0.18 ± 0.02	31
30	171.92 ± 36.64	26.15 ± 0.48	18.60 ± 0.16	13.97 ± 3.64	0.19 ± 0.052	21
40	123.45 ± 19.18	18.30 ± 1.08	15.17 ± 0.18	8.67 ± 5.54	0.21 ± 0.018	14
50	94.82 ± 42.38	16.27 ± 2.37	11.52 ± 0.92	8.70 ± 4.38	0.45 ± 0.069	10
60	61.01 ± 25.050	11.37 ± 0.99	10.11 ± 0.58	7.2 ± 3.41	0.50 ± 0.028	8

5.4.5 Collaborative-Competitive Scenarios

Collaborative vs. Competitive

Until now, all experiments were carried out assuming a fully cooperative game, where all agents share the same networks (actor and critics), episode buffer and the reward function. We explored the performance of the proposed method in a competitive scenario, i.e., where agents seek to maximise their own individual rewards. For this, we created a separate actor and critic network for each agent and allowed them to maximise their own speed and penalise their collisions. Having a separate network for each agent makes the training substantially less efficient, increasing the training time nearly threefold. For this set of experiments, we used the two-lane road scenario and trained the MAPPO algorithm for 2000 episodes.

In both the cooperative and competitive scenario, the MAPPO algorithm can achieve low collision rates and high safety distances between all vehicles. In addition, the resulting policies allow the EMV to go at a faster average speed. However, in a fully competitive scenario, vehicles do not coordinate to let the EMV pass efficiently. Figure 5.11 contrasts the difference between cooperative and competitive multi-agent policies. In the fully cooperative case, vehicles move aside to minimise the braking and the number of lane changes of the EMV. In contrast, in the

competitive scenario, the AVs do not change lanes preemptively, and they just maximise their own velocity; therefore, the EMV has to constantly overtake other vehicles. This causes the EMV to achieve a considerably lower average speed, which can be seen in figure 5.12 (a). On the other hand, figure 5.12 (b) shows the difference in average risk of collision for the EMV. Due to the constant overtaking manoeuvres done by the EMV, the risk of collision of the substantially greater in the competitive case. The MAPPO algorithm can still generate collision-free policies in this case but the overall traffic efficiency is substantially lower.

Different Levels of Collaboration

The performance of the proposed method on different levels of collaboration was evaluated through a series of experiments, wherein we varied the collaborative reward component c in equation 5.8. We created four scenarios, varying the reward component from 1 to 0. Table 5.5 presents the results of these experiments, which demonstrate that decreasing the collaboration component leads to lower levels of EMV speed and higher collision risks. Furthermore, the EMV vehicle had to perform a greater number of lane changes as the AVs did not yield to it. It is noteworthy that as the collaboration factor was reduced, all vehicles tended to move at lower speeds and closer to each other.

Table 5.5: Results of the experiments for different levels of collaboration. The scenarios are represented by different values for the cooperative reward component c.

Collaboration Factor c	EMV Speed (m/s)	AV Speed (m/s)	Col. rate	Col. risk	Safety Dist. (m)	Avg. EMV LCM
1	26.45 ± 2.77	17.45 ± 1.25	2.67 ± 1.14	0.184 ± 0.081	17.22 ± 5.64	39.26
0.75	22.25 ± 2.35	18.55 ± 2.12	3.97 ± 2.11	0.178 ± 0.092	17.01 ± 6.37	37.16
0.25	19.23 ± 3.67	15.53 ± 2.86	6.14 ± 3.21	0.268 ± 0.121	14.32 ± 3.83	47.52
0	16.48 ± 2.16	12.45 ± 2.01	9.56 ± 4.35	0.301 ± 0.486	12.34 ± 3.35	58.12

Figure 5.13 presents two contrasting scenarios: fully cooperative (left) and fully competitive (right). On the left, in the fully cooperative scenario, the AVs collaborate effectively, allowing them to maintain a high speed of around 20 m/s while the EMV travels at an even higher speed of 25-26 m/s. This cooperative behaviour ensures smooth coordination and high safety distances between vehicles, facilitating rapid and efficient movement for both the AVs and the EMV.

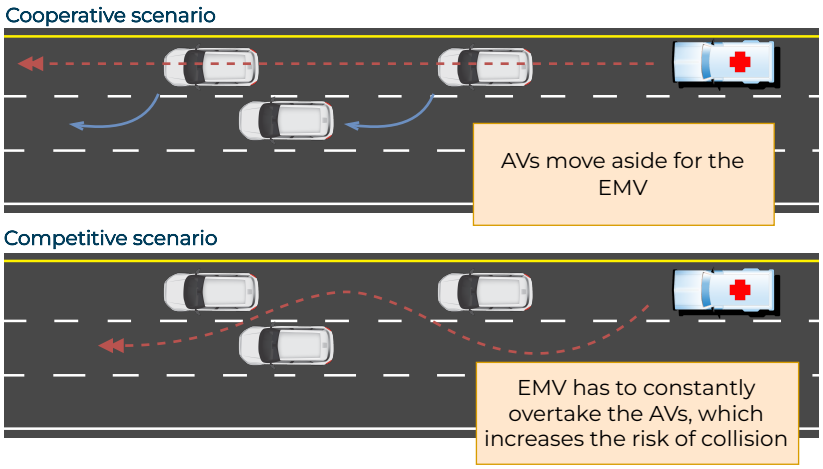
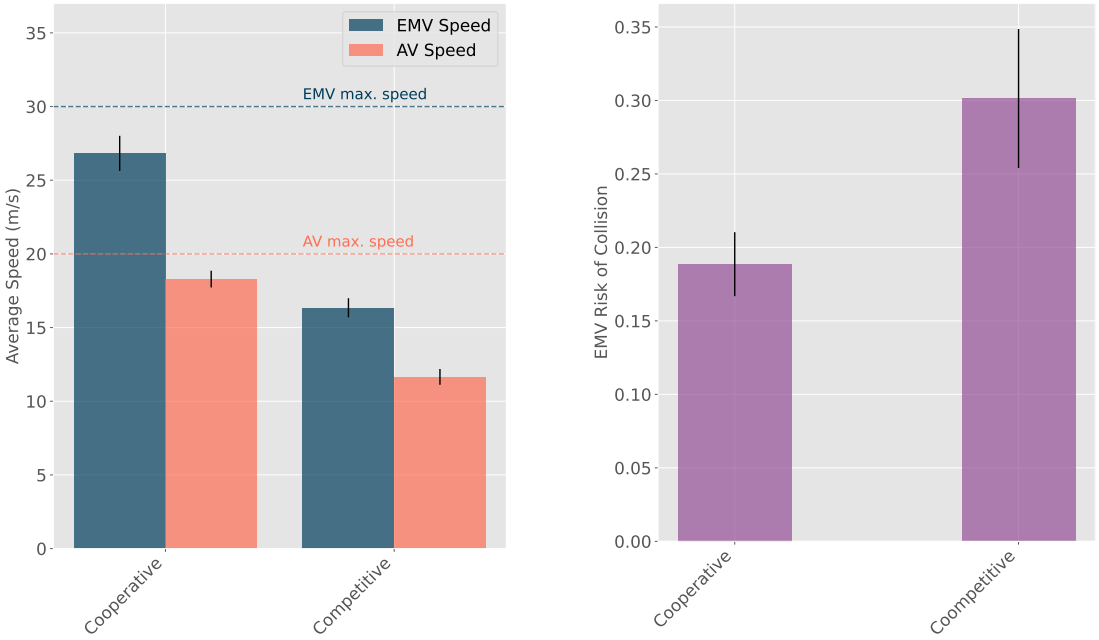


Figure 5.11: Cooperative vs. competitive multi-agent policies. In the fully cooperative scenario, the AVs move aside to let the EMV pass, while in the competitive scenario the EMV has to constantly perform overtaking manoeuvres.

On the right, the fully competitive scenario leads to more conservative behaviour from the AVs, as they are focused on individual objectives rather than cooperation. As a result, the AVs navigate at lower speeds, around 15 m/s, and the EMV must also reduce its speed to around 17-19 m/s, as the AVs do not give way. This highlights the challenges in ensuring smooth transit for the EMV in non-cooperative environments, where AVs prioritize their own safety, leading to slower overall movement and increased collision risk.

5.5 Concluding Remarks

AVs have the potential to improve road safety and efficiency, although there are still many challenges regarding their practical application in traffic systems. This chapter proposed a multi-agent approach for the safe and efficient manoeuvring of autonomous traffic in the presence of EMVs. We used the MAPPO algorithm combined with risk metrics to determine manoeuvring actions for all cars in congested road settings. We explored the capabilities of the proposed approach in five case studies. First, we compared it with a traditional rule-based car following model. Second, we contrasted the two main components of the reward function: risk and traffic efficiency. Third, we investigated the scalability and performance of different num-



(a) (b)

Figure 5.12: Comparison between the cooperative and competitive scenarios: (a) Average speeds (b) EMV Risk of collision

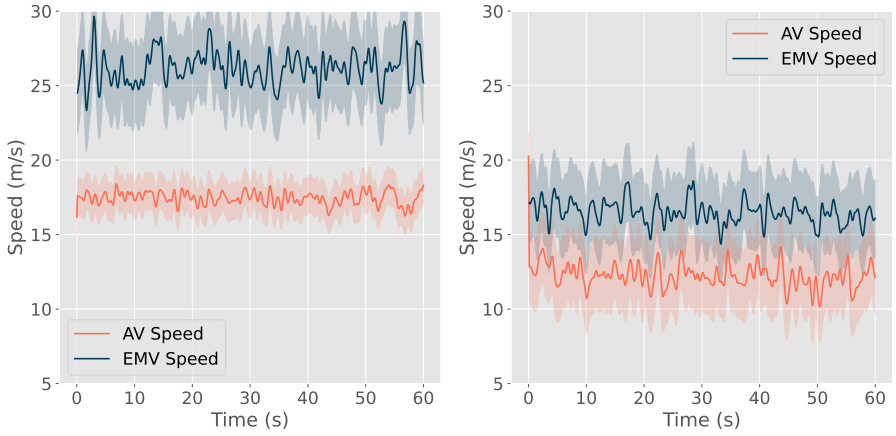


Figure 5.13: Speed Profiles in Fully Cooperative vs Fully Competitive Scenarios. The left plot represents a fully cooperative setting with AVs coordinating effectively, while the right plot shows a competitive setting where AVs operate independently. The speed of the Autonomous Vehicles (AVs) and the Emergency Vehicle (EMV) are depicted.

bers of agents. Finally, we presented a competitive scenario where vehicles seek to maximise their own independent reward.

The results of the five case studies show the potential and capabilities of the proposed method. It consistently outperforms the car following baselines, obtaining higher average speeds for the EMV while maintaining a lower overall risk of collision. Furthermore, we show how the

reward function can be tuned to achieve higher safety or traffic efficiency. However, a minimum amount of risk component is needed to guarantee safe transit and no collisions. A higher number of agents translates into lower average speeds of the EMV and AVs; however, the algorithm can still guarantee safe manoeuvring. The proposed method is also applicable to competitive scenarios, where vehicles act selfishly, maximising their own speed. The resulting policies involve constant manoeuvring of the EMV, which increases its risk of collision when compared to a fully cooperative case.

There is some future work that can explore new settings and overcome some of the limitations of this study. We considered just a single road environment, with vehicles going in one direction. Other scenarios can be evaluated, such as highway merging, two-directional lanes, intersections, and the possibility of the EMV moving in the contraflow. In addition, explicit communication between agents to improve overall traffic safety should be explored in the future. However, having a one-to-one communication network between CAVs is not scalable; thus, innovative, efficient methods are needed for cases with multiple agents. The next chapter will develop methods for efficient communication in the context of these challenging scenarios.

Chapter 6

Communication Strategies for Coordinated Autonomous Driving

This chapter has been adapted from the following publication:

L. Parada, H. Tian, J. Escribano, and P. Angeloudis. An End-to-End Collaborative Learning Approach for Connected Autonomous Vehicles in Occluded Scenarios. In 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), pages 5548–5554, Sept. 2023. ISSN: 2153-0017.

6.1 Introduction

In the previous chapter, we demonstrated how Autonomous Vehicles (AVs) can achieve high levels of coordination in challenging emergency vehicle scenarios, even without relying on direct vehicle-to-vehicle communication. However, in some situations, such as low-visibility scenarios or complex urban intersections, communication becomes essential for ensuring safety and efficiency.

Within the multi-level decision-making framework established in Chapter 1 (see Figure 6.1), communication is beneficial at multiple decision-making levels, but it becomes especially critical at the tactical level, where AVs must make coordinated decisions in dynamic and uncertain environments. At this level, local observations alone are often insufficient, and communication enables AVs to share crucial information such as perception data and intended manoeuvres.

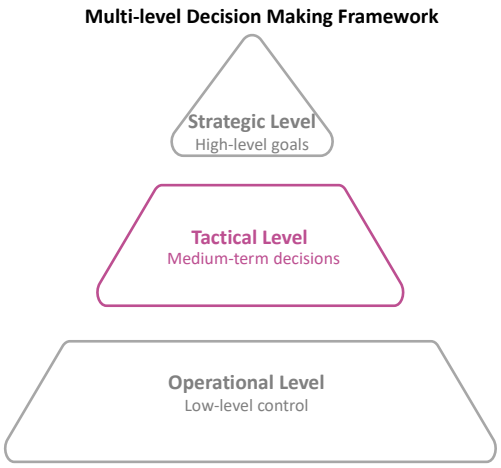


Figure 6.1: Communication plays a vital role at the tactical decision-making level for improved coordination and safety.

As discussed in the Literature Review chapter, recent advances in collaborative perception have shown that sharing sensory data between vehicles via Vehicle-to-Vehicle (V2V) communication can significantly extend an AV's situational awareness, particularly in occluded or low-visibility environments. This enhanced perception capability is crucial for preventing accidents in urban settings, where hidden obstacles or rapidly approaching vehicles may pose serious safety risks (Cui et al., 2022). However, transmitting large volumes of raw sensor data introduces substantial computational and bandwidth challenges. To address these limitations, various data fusion strategies have been developed. Among these, intermediate fusion has emerged as a promising approach, offering a balance between bandwidth efficiency and perception accuracy by transmitting processed data rather than raw inputs (Xu et al., 2022a; Wang et al., 2020b; Xu et al., 2022b).

Nevertheless, the scalability of these methods poses a significant challenge, especially as the number of connected vehicles grows. To manage the increasing communication demands, selective communication strategies such as Who2Com (Liu et al., 2020b) and When2Com (Liu et al., 2020a) have been introduced. These techniques dynamically prioritise which data to share and with which vehicles, thus optimising bandwidth usage while maintaining high levels of performance and safety.

Despite these advancements, the existing literature on communicative MARL and collaborative perception for AV coordination remains focused on isolated components. While these approaches improve the ability of CAVs to share environmental knowledge and coordinate actions, especially in complex or low-visibility scenarios, few studies address the problem from an end-to-end perspective, encompassing both perception and control. Most research has concentrated on perception or perception and prediction, often relying on supervised learning approaches like behaviour cloning methods, which are prone to issues such as data biases, causal confusion, and high variance (Cui et al., 2022; Codevilla et al., 2019). Furthermore, acquiring expert data, which these approaches depend on, is often impractical, particularly for rare or edge cases. This chapter seeks to bridge this gap by investigating how the integration of MARL with V2V communication can provide a more robust, end-to-end solution—improving not only perception but also decision-making and control.

This chapter is divided into two sections, presenting two different approaches to AV communication: Section 6.2 presents a MARL framework designed to enhance safety through the strategic transmission of vehicle intent. Section 6.3 presents a practical application of V2V communication in occluded driving scenarios. The primary focus is on coordination among CAVs, with HDV interactions briefly included to demonstrate the potential for mixed-traffic environments. Together, these sections illustrate the critical role of communication in enhancing the collaborative capabilities of autonomous vehicles, ensuring that they can operate safely and efficiently, even in the most challenging conditions.

6.2 Improving Safety by Communicating Vehicle Intent

Agent coordination has been extensively studied within the MARL community by sharing state representations through agent communication networks (Zhu et al., 2022). However, we argue that for safety-critical applications such as autonomous driving–characterised by highly dynamic scenarios–simply sharing state features is not enough. Understanding the future intentions of others is crucial for accurate decision-making. Although some studies have explored

intention-sharing in MARL (Jiang and Lu, 2018; Chu et al., 2020), their applications are limited to simplistic grid-like environments, lacking the intricacies of autonomous driving, such as high-dimensional observations (e.g., RGB images, LiDAR) and autonomous traffic dynamics.

To address the existing gap, we introduce an intention-sharing MARL framework tailored for CAVs to navigate complex urban environments safely and efficiently. Our framework orchestrates two communication stages among agents. In the first stage, agents share encoded messages and their prior policy outputs. Subsequently, in the second stage, they communicate their anticipated intent across a future horizon H. Each agent employs an observation predictor module, enabling the prediction of forthcoming environmental states. Using these predictions, agents project their future actions over the designated time horizon H by executing a policy rollout. To efficiently manage this process, agents transmit their encoded observations and intended actions using a scheduler for bandwidth optimisation and a graph attention network for message processing. The overall architecture employs an actor-critic loss alongside an independent supervised loss for training the observation prediction module. Training occurs directly from environmental interaction, eliminating the need for pre-training for the observation predictor. We benchmark our method against state-of-the-art Communicative MARL approaches across diverse urban scenarios and show how it can efficiently learn safer policies with lower communication graph density.

6.2.1 Framework

We formulate the communicative MARL problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), represented by the tuple $\langle N, S, A, O, \mathcal{T}, R, \gamma, \mathcal{M}, \mathcal{C} \rangle$. Here, $N = \{1, 2, \ldots, n\}$ is the number of agents. S denotes all possible states, where $s^t \in S$ represents the environment's state at time t. The set A comprises all possible actions, with each agent i selecting an action $a_i^t \in A$ at each time step.

Agents receive partial information via observations $o_i^t \in O$, and the transition between states, given agents' actions, is defined by the transition function $\mathcal{T} : S \times A \to S$. The reward function

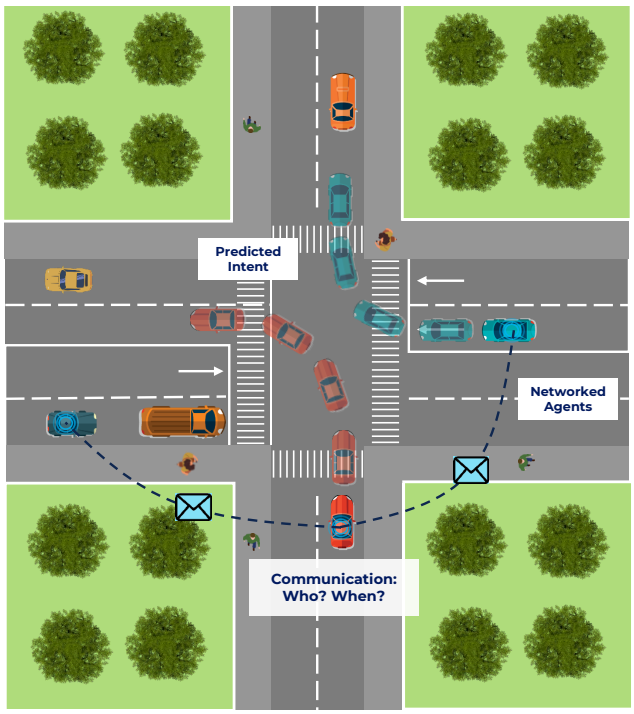


Figure 6.2: Illustration of our MARL framework in an urban traffic context, showcasing the strategic exchange of state information and vehicle intent, coordinated by a scheduler to optimise bandwidth and enable real-time decision-making for autonomous vehicles.

$R : S \times A \times S \to \mathbb{R}$ determines the immediate reward, balancing current and future rewards through the discount factor $\gamma \in [0, 1]$.

Communication occurs through messages $\mathcal{M}$ and is governed by the protocol $\mathcal{C} : \mathcal{M}^N \to \mathcal{M}$, which defines how messages are aggregated. The overall objective is to determine policies $\pi_i : O \times \mathcal{M} \to A$ for each agent that maximize the expected cumulative reward:

$$\max_{\pi_1, \pi_2, \ldots, \pi_N} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r^t(s^t, a_1^t, a_2^t, \ldots, a_N^t, s^{t+1})\right], \tag{6.1}$$

where $r^t(s^t, a^t, s^{t+1})$ is the immediate reward at time t, defined by $R(s^t, a^t, s^{t+1})$.

Our proposed Intention-sharing MARL framework, as illustrated in Figure 6.3, is designed to facilitate communication among agents in a cooperative setting using a communicative actor-critic model. This framework operates in two key communication stages, ensuring that agents can effectively share and utilize information for coordinated decision-making.

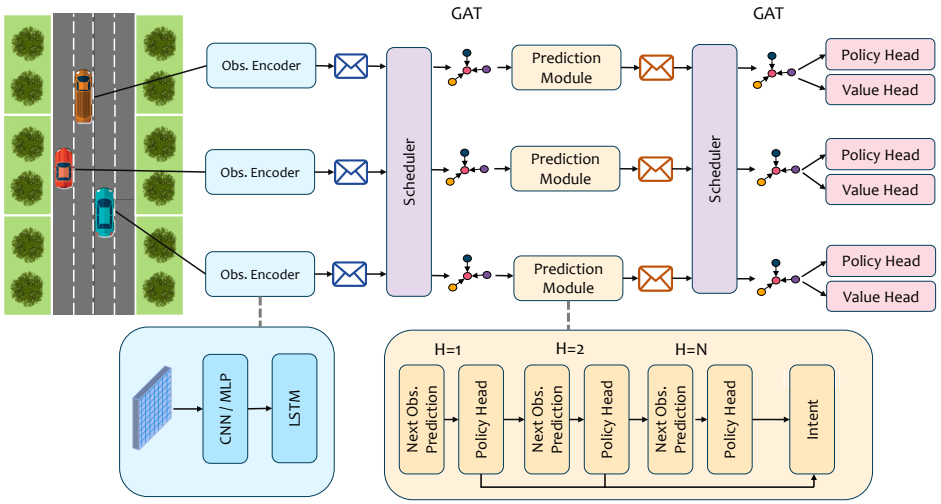


Figure 6.3: This figure depicts IntNet, our MARL framework for safe, efficient CAV navigation, highlighting the sharing of encoded observations and vehicle intentions. The scheduler optimizes the communication graph, while the GAT module processes incoming messages with attention to prioritize information for decision-making.

Stage 1: Observation Encoding and Policy Fingerprint Sharing. In the first stage, each agent i encodes its local observation o_i^t at time step t into a message m_i^t using an encoding function $f_{\text{enc}} : O \to \mathcal{M}$. The encoded message $m_i^t = f_{\text{enc}}(o_i^t)$ effectively summarizes the observed state information that can be shared with other agents. Alongside this, each agent also broadcasts its policy fingerprint from the previous time step, denoted by $\mathbf{fp}_i^{t-1}$. The policy fingerprint is defined as:

$$\mathbf{fp}_i^{t-1} = \pi_\theta(o_i^{t-1}, m_{-i}^{t-1}), \tag{6.2}$$

where $\pi_\theta : O \times \mathcal{M} \to A$ is the policy function parameterized by θ, which maps the agent's observation o_i^{t-1} and the aggregated messages from all other agents m_{-i}^{t-1} to an action a_i^{t-1}. The subscript $-i$ indicates the set of all agents except agent i. This process enables agents to share both their current observations and their recent decision-making context, establishing a foundation for collaborative action planning.

Stage 2: Intention Prediction and Sharing. In the second stage, each agent predicts and shares its future intentions over a specified time horizon H. These predictions are represented by a sequence of future policy fingerprints $\mathbf{I}_i^{t+H}$, which includes the agent's projected actions

and observations for the upcoming H steps. The intention set $\mathbf{I}_i^{t+H}$ is formulated as:

$$\mathbf{I}_i^{t+H} = \begin{bmatrix} \mathbf{fp}_i^{t+H} \\ \mathbf{fp}_i^{t+H-1} \\ \vdots \\ \mathbf{fp}_i^{t+1} \end{bmatrix} \tag{6.3}$$

where each future policy fingerprint $\mathbf{fp}_i^{t+h}$ for $h = 1, 2, \ldots, H$ is computed as:

$$\mathbf{fp}_i^{t+h} = \pi_\theta \left(\hat{\mathbf{o}}_i^{t+h}, m_{-i}^{t+h} \right). \tag{6.4}$$

In this equation, $\hat{\mathbf{o}}_i^{t+h}$ represents the predicted observation of agent i at time step $t+h$, which is derived using the observation prediction module $f_{\text{obs}} : O \times \mathcal{M} \to O$. The predicted observation $\hat{\mathbf{o}}_i^{t+h}$ is calculated recursively starting from the agent's current observation o_i^t and the policy fingerprints from the other agents. The recursive prediction is defined as:

$$\hat{\mathbf{o}}_i^{t+h} = \begin{cases} f_{\text{obs}} \left(\mathbf{o}_i^t, \mathbf{fp}_{-i}^{t-1} \right) & \text{if} \quad h = 1 \\ f_{\text{obs}} \left(\hat{\mathbf{o}}_i^{t+h-1}, \mathbf{fp}_{-i}^{t+h-1} \right) & \text{if} \quad h > 1 \end{cases} \tag{6.5}$$

where $\mathbf{fp}_{-i}^{t+h-1}$ denotes the collective policy fingerprints from all agents except i at time step $t + h - 1$. This recursive mechanism allows each agent to anticipate future states and actions, forming a detailed plan of intended actions over the horizon H.

Communication Scheduling and Graph Attention. Given limited bandwidth constraints and the high-dimensional state space inherent in autonomous driving, our approach incorporates a scheduler module for each communication stage. This scheduler precisely dictates who and when to communicate in each step. For every round of communication, it establishes the adjacency matrix G^t, which specifies the connections between communicating entities, thereby optimizing both the flow and relevance of information shared:

$$G^t = \begin{bmatrix} G^t_{11} & G^t_{12} & \cdots & G^t_{1N} \\ G^t_{21} & G^t_{22} & \cdots & G^t_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ G^t_{N1} & G^t_{N2} & \cdots & G^t_{NN} \end{bmatrix} \tag{6.6}$$

where $G^t_{ij} = 1$ when agent i communicates with agent j at time step t, and $G^t_{ij} = 0$ otherwise. By training the whole pipeline we can search for the optimal policy while reducing bandwidth utilisation.

Agents use a Graph Attention Network (GAT) following each communication stage to merge and process the received messages. The GAT employs an attention mechanism to dynamically weigh the importance of each message, enabling agents to focus on the most relevant information for decision-making. This process is mathematically represented as follows. Given a node i with feature representation h_i and its neighbors $\mathcal{N}(i)$, the attention coefficient α_{ij} between node i and its neighbor j is computed as:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_j]\right)\right)}{\sum_{k \in N(i)\cup\{i\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_k]\right)\right)} \tag{6.7}$$

where $\|$ denotes concatenation, $\mathbf{W}$ is a weight matrix applied to each node, $\mathbf{a}$ is a weight vector for the attention mechanism, and LeakyReLU is a nonlinear activation function. The attention coefficients α_{ij} are normalized across all neighbours k of i using the softmax function, ensuring comparability.

The updated feature representation h'_i of node i, incorporating information from its neighbours, is then calculated as a weighted sum:

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)\cup\{i\}} \alpha_{ij}\mathbf{W}h_j\right) \tag{6.8}$$

where σ denotes an optional activation function, this formulation allows the GAT to adaptively prioritize information from the most relevant neighbors, efficiently integrating transmit-

ted messages into meaningful features for enhanced decision-making. Finally, once the agents have processed the incoming messages, they determine the policy and value outputs through independent linear layers as depicted in figure 6.3.

Training

To train the observation prediction module f_{obs}, we collect (input, output) pairs from episode rollouts as described by Nagabandi et al. (2017). In contrast to their method, we do not pre-train the predictor and instead only rely on transitions stored during the episode rollout. Each episode step generates transitions $\tau = (o^t, \pi^0, o^{t+1})$, from which we compile a supervised dataset with inputs o^t, a^t and targets o^{t+1}. Observations are normalised, and variables are categorised as continuous or categorical. To train f_{obs}, we employ the Mean Squared Error (MSE) loss:

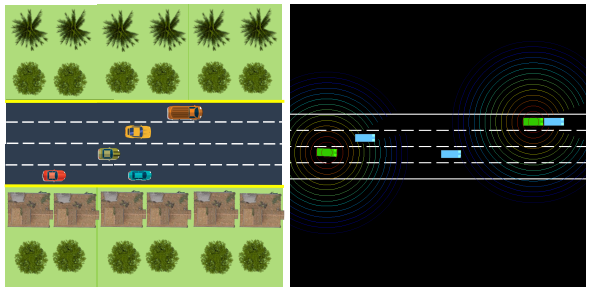
$$\mathcal{L}_{obs} = \frac{1}{N} \sum_{i=1}^{N} (o_i^{t+1} - \hat{o}_i^{t+1})^2, \tag{6.9}$$

where o_i^{t+1} represents the actual next observation, $\hat{o}_i^{t+1}$ denotes the predicted next observation, and N is the total number of samples.

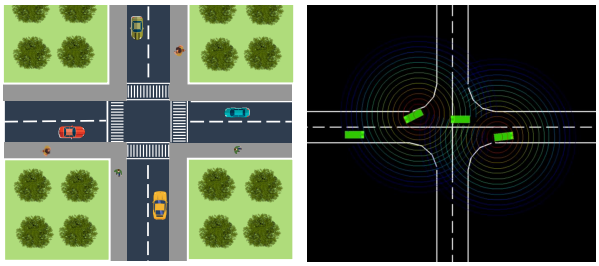
To train the policy, we use an actor-critic loss, which incorporates the loss of the actor (policy) as well as the loss critic (value):

$$\begin{aligned} \mathcal{L}_{AC} =& \mathbb{E}_{(\tau)} \left[\sum_{t=0}^{T} A^t \cdot \nabla_\theta \log \pi_i(a_i^t | o_i^t, m_i^t) \right] \\ &+ \lambda \sum_{t=0}^{T} \nabla_\rho (R_i^t - V_\rho(o_i^t))^2 \end{aligned} \tag{6.10}$$

where $A^t = R_i^t - V_\rho(o_i^t)$ is the advantage function, and $V_\rho(o_i^t)$ is the value function.



(a) Highway.



(b) Intersection.

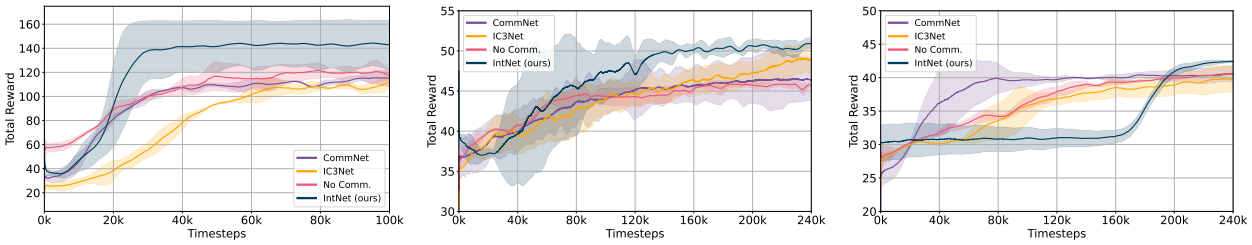


(c) Lane merging.

Figure 6.4: Scenarios used for the experiments.

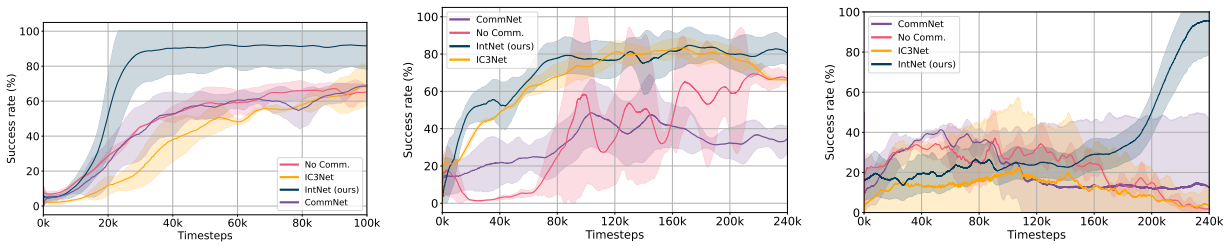
Environment Description

We use a modified version of Highway-env (Leurent, 2018), adapted for multi-agent urban navigation where agents avoid collisions and reach their destination to maximise rewards. Our approach introduces randomness in vehicles' speeds, locations, and destinations to enhance adaptability, and limits agent communication to 60 meters to maintain realism, aligning with DSRC protocols.



(a) Highway. (b) Intersection (c) Lane Merging

Figure 6.5: Team reward of the different methods during training.



(a) Highway. (b) Intersection (c) Lane Merging

Figure 6.6: Success rate of the different methods during training.

The elements of the gym environment can be summarised as follows:

- Agents: CAVs with limited visibility, capable of exchanging messages through a vehicle-to-vehicle communication network.
- Observation: A bespoke LiDAR-like method using raycasting to segment the 360° field around the agent, detecting other vehicles within an 80-meter range.
- Action: Discrete meta-actions including acceleration (minor and major), deceleration (minor and major), speed maintenance, and lane changes (left and right).
- Reward: A cooperative game where rewards are based on speed, collision avoidance, arrival, and bonuses for cooperative lane merging.

6.2.2 Experiments

We benchmark our method against baselines across three scenarios: Highway, 4-way Intersection, and Lane Merging, depicted in Figure 6.4, which includes both schematic and actual environment representations to illustrate the observations. The Highway and Lane Merging scenarios include both CAVs and human-driven vehicles (HDVs), while the Intersection scenario is limited to CAVs only. Though the focus is on achieving coordination among CAVs, HDVs are included in some scenarios to increase realism. HDVs are modelled using the IDM/MOBIL car-following model (Kesting et al., 2007), with parameters sourced directly from this paper, where they were calibrated with macroscopic traffic data. This inclusion allows us to benchmark CAV coordination within a more realistic mixed-traffic setting.

The primary evaluation metrics in our study are the average arrival and collision rates, expressed in percentages. The arrival rate quantifies the percentage of episodes where all CAVs successfully reached their destinations, while the collision rate measures the percentage of episodes involving at least one CAV collision. Additionally, the average team return is evaluated to assess overall performance. To ascertain the robustness of our training outcomes, we utilise three independent seeds to generate learning curves and validate our results over 100 test episodes.

Baseline comparison

We evaluated our method's performance against established communication-based MARL approaches, which primarily focus on sharing state or state features. The comparison included three baseline methods: No Communication, CommNet (Sukhbaatar et al., 2016), and IC3Net (Singh et al., 2018).

Figures 6.5 and 6.6 illustrate the team reward and arrival statistics, respectively, across three scenarios for all methods, with IntNet consistently achieving higher rewards and success rates. Variations in arrival rates indicate that intention-sharing is particularly beneficial in specific scenarios, such as lane merging, which is crucial for safe navigation, while its impact in the intersection scenario is smaller. Table 6.1 further details the arrival rate (%) test performance over 100 independent episodes in these scenarios, supporting these observations.

In the highway scenario, our proposed method enables agents to reach high success levels with just 100k training episodes, compared to baselines, which require around 200k episodes for similar success and safety outcomes. In the intersection scenario, although IC3Net shows success comparable to our IntNet method, ours provides a marginally quicker and more consistent learning curve. We see that all baseline algorithms underperform in the lane-merging scenario. This scenario demands the most cooperation out of the three, causing baseline algorithms to learn inconsistently and often favour solutions that increase the speed at the expense of safety, leading to more collisions.

Figure 6.7 showcases IntNet's training progress in the Highway scenario, where the rapid decline

Table 6.1: Test arrival rate (%) of different Communicative MARL methods.

Method	Highway	Intersection	Lane Merging
No Comm	62.6 ± 3.6	65.3 ± 8.4	7.7 ± 4.3
IC3Net	59.5 ± 6.1	70.7 ± 6.3	12.3 ± 7.1
CommNet	64.1 ± 8.4	41.0 ± 8.2	28.4 ± 14.2
IntNet (Ours)	97.1 ± 2.6	89.7 ± 5.9	92.8 ± 9.9

in observation prediction loss validates the efficacy of our training methodology, achieved in a model-free setting without pretraining. The swift reduction in collision rate reflects the model's quick convergence, while the increased average vehicle speed indicates the development of efficient policies by the model.

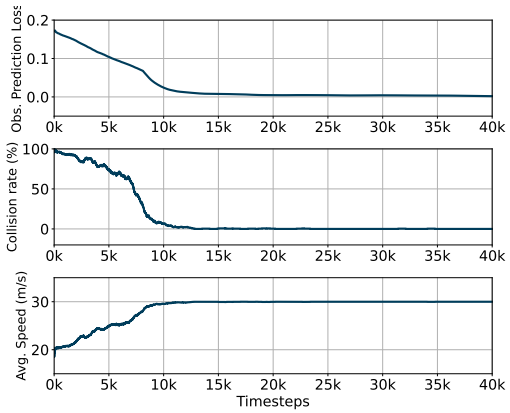
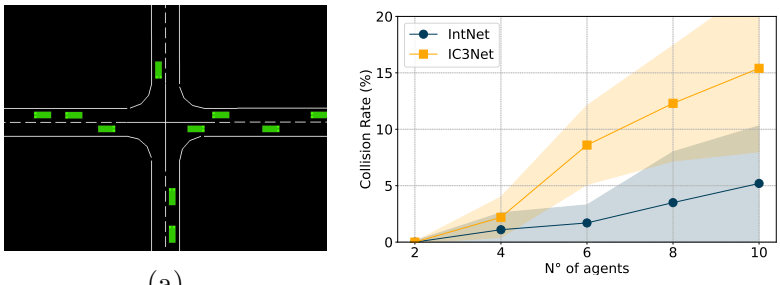


Figure 6.7: Evolution of statistics for IntNet during training on the Highway scenario.

Scalability

To assess the scalability of our methodology, we performed experiments in an intersection scenario, progressively increasing the number of agents within the network. The results, depicted in Figure 6.8 and Table 6.2, show the test collision rates (%) as the agent count varies. As expected, the environment's complexity escalates with each additional agent, making it increasingly difficult to maintain a zero collision rate. Nevertheless, our IntNet approach produces significantly lower collision rates compared to the IC3Net model when the number of agents is increased. The IC3Net model exhibits a growing number of collisions as more agents are added, with rates climbing over 15% when the environment includes ten agents, highlighting the challenges of scaling in more congested settings, even with communication.



(a) (b)

Figure 6.8: (a) Intersection scenario with ten agents. (b) Effect of increasing the number of agents on the collision rate.

Table 6.2: Test collision rates (%) for IntNet and IC3Net across various agent counts.

Agent Count	IntNet	IC3Net
2	0.0 ± 0.12	0.0 ± 0.05
4	1.1 ± 1.5	2.2 ± 1.8
6	1.7 ± 1.6	8.6 ± 3.5
8	3.5 ± 4.5	12.3 ± 5.12
10	5.2 ± 5.1	15.4 ± 7.4

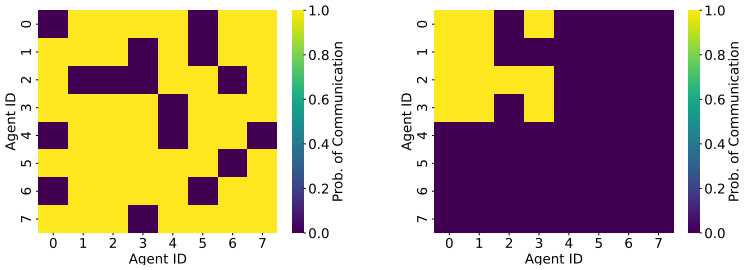
Communication Efficiency

Our architecture includes a scheduler component that optimises message transmission, decreasing the communication network's load. We evaluated its impact through an experiment in an intersection scenario solely with CAVs, contrasting network performance with and without the scheduler. Without the scheduler, the system defaults to a fully connected network, facilitating communication among all agents within range. Table 6.3 compares network performance in the intersection environment with the scheduler, revealing that each agent communicates with fewer than half of the agents, on average, while achieving the same success level.

Agents communicate only at crucial times for safe navigation, illustrated by Figure 6.9, showing the communication graph in the intersection environment at two key moments. Figure 6.9a reveals the graph as agents approach the intersection, and Figure 6.9b displays it as most agents exit, with links in 6.9b (vehicle IDs 0 to 3) indicating vehicles that are still in the intersection.

Table 6.3: Effect of the communication scheduler on graph density and success rate.

	Graph Density	Success rate (%)
No Scheduler	1.0 ± 0.0	86.3 ± 5.1
Scheduler	0.41 ± 0.4	88.2 ± 5.8



(a) Entering the intersection. (b) Leaving the intersection

Figure 6.9: Heatmaps of the communication graphs when entering and leaving the intersection.

6.2.3 Summary

This section introduced a novel method that shares vehicle intent, addressing the critical challenge of enhancing coordination and safety among autonomous vehicles. We accomplish this by incorporating an observation prediction module into a dual-stage communicative actor-critic architecture. Our model-free training approach uses transitions recorded during episode rollouts to simultaneously train the observation predictor and the policy. The results demonstrate the efficacy of our intent-sharing strategy in fostering safer and more efficient policies in complex urban settings with multiple agents while minimising the communication graph density. Our findings reveal that sharing intent not only enhances safety and efficiency but can also accelerate learning convergence, streamlining the path to optimal policies in complex multi-agent environments.

6.3 Safe Navigation of Occluded Intersections

This section addresses more challenging scenarios that involve visual occlusion, i.e., situations where objects block the line of sight of autonomous vehicles. Consider a scenario such as the one presented in Figure 6.10, where an obstructed vehicle in one lane becomes invisible to others,

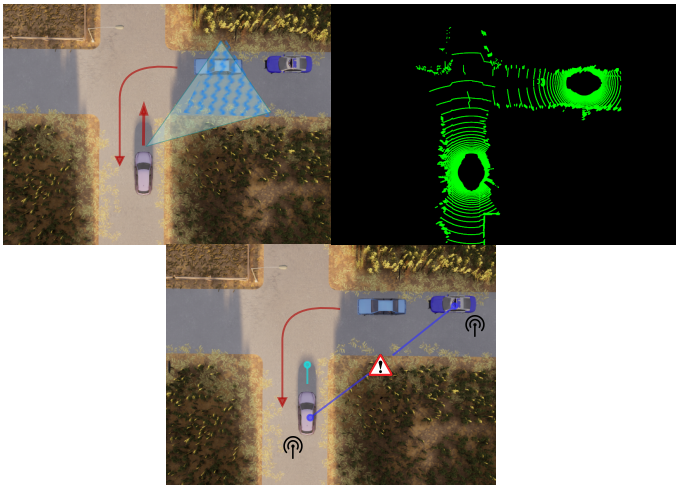


Figure 6.10: Collaborative CAV control. Some vehicles may be occluded by tall trees/buildings or other vehicles in the scene. With the use of LiDAR features from nearby agents, safe and efficient policies can be developed to avoid accidents.

emphasising the need for collaborative perception to avoid a potential collision. Collaborative perception allows CAVs to exchange information via a V2V ad-hoc network (Tonguz et al., 2007), ensuring real-time sharing of perception data. However, sharing raw data in real-time creates computational and bandwidth challenges, prompting the need for efficient techniques aligned with hardware limitations.

We propose an end-to-end collaborative learning method to navigate dangerous scenarios involving occlusion. The method is based on a decentralised perception mechanism that preprocesses raw LiDAR data, which are then shared with nearby agents using an ad-hoc V2V network. The features extracted from LiDAR are received by nearby CAVs and utilised within a multi-agent reinforcement learning pipeline to obtain the desired driving action. Specifically, we use Multi-agent Proximal Policy Optimisation, a MARL method that has been shown to achieve superior performance in several cooperative games (Yu et al., 2021). In order to train our method, we design a gym environment based on the CARLA simulator that generates challenging occluded intersections with two or more CAVs and multiple human-driven vehicles.

6.3.1 Framework

Occluded Intersection Gym Environment

There are several autonomous driving gym environments available for evaluating DRL algorithms, such as SMARTS (Zhou et al., 2020b), Highway-env (Leurent, 2018), and MACAD (Palanisamy, 2019b). However, none of these environments satisfied our requirements for cooperative driving capabilities and realistic LiDAR point cloud representations. While some environments, such as SMARTS and MACAD, provide multi-agent capabilities, they do not have realistic LiDAR point cloud representations. On the other hand, Highway-env has realistic LiDAR point cloud representations but lacks cooperative driving capabilities. To address these limitations, we developed our own environment called the Occluded Intersection Gym Environment.

Our proposed gym environment utilises CARLA (Dupuis et al., 2010) and the OpenCDA framework (Xu et al., 2021) to simulate challenging cooperative scenarios in an episodic manner. The environment includes two or more CAVs and traffic vehicles in high-risk scenarios that may result in collisions. The CAVs can share information and take independent actions in real-time while traffic vehicles follow predetermined routes established by the SUMO traffic simulator, without sharing information or taking any actions. We also introduce randomisation in specific simulation parameters, such as the types and characteristics of CAVs and traffic vehicles and their initial poses, so that we can train on different variations of the same scenario.

In each episode of our proposed gym environment, traffic vehicles follow one of several predefined routes through the intersection, all leading to challenging occluded scenarios. This presents a realistic and diverse set of scenarios for the CAVs to navigate safely and efficiently. To aid in this navigation, CAVs are able to share both metadata (such as pose and speed) and LiDAR features with nearby agents in real-time. We limit this sharing to agents within a predefined range, in order to maintain realism and avoid unrealistic global knowledge.

The elements of the gym environment can be summarised as follows:

- Agents: The agents correspond to the CAVs, which have partial observability of the environment and can share real-time information with other agents.

- Observation: Includes metadata from the ego vehicle and vehicles within a specific range. This data consists of the current speed, vehicle dimensions and current lane id. The raw LiDAR data consists of 4D point clouds, where the first three dimensions correspond to the coordinates of each point (x, y, z), and the last dimension represents the intensity loss during travel.

- Action: The action space consists of five discrete values: two for speeding up (smaller and larger amounts), two for slowing down (smaller and larger amount) and one for doing nothing.

- Reward: We assume a fully cooperative game; thus, agents share the same reward, defined by the following equation:

$$R = -5 * r_{col} + r_{speed} + 5 * r_{dest} + -0.01 * r_{step} \tag{6.11}$$

  Where r_{col} is a collision penalty, r_{speed} is a positive reward for the speed of the vehicle, r_{dest} is the reward given when the CAV reaches its desired destination and r_{step} is a minor penalty given to encourage the agent to reach the goal promptly.

Communication Message

The V2V network enables communication among vehicles using compressed feature messages that encapsulate scene information from each CAV's perspective. We denote M_i as the message shared by CAV i, given by $M_i = CNN(\hat{z}_i^{(0)})$, where CNN comprises a sequence of convolutional layers and $\hat{z}_i$ represents the preprocessed LiDAR point cloud data.

After testing various configurations, we found that utilising two convolutional layers yielded the best results while considering bandwidth limitations, which will be discussed in the Experiments section. The resulting feature size of the output is (4, 128, 128). It's important to

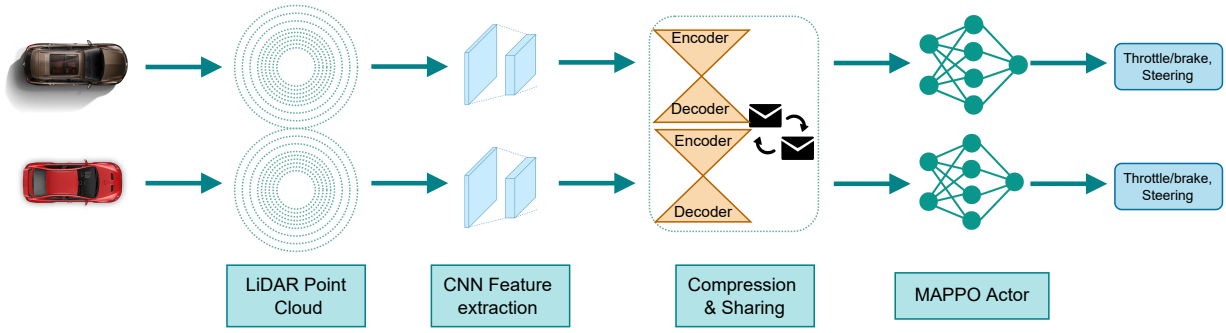


Figure 6.11: Collaborative MARL framework. LiDAR features are extracted from preprocessed point cloud data and shared through a V2V local communication network. The aggregated messages are then used to calculate the desired action through the actor network.

acknowledge that our current message format is a custom design designed to maximise message information while complying with bandwidth limitations. Future efforts should focus on developing a message format that aligns with established standards, such as the ETSI standard (ETSI, 2014).

Collaborative MARL framework

To ensure safe and efficient navigation through occluded intersections, we developed a collaborative framework based on the MAPPO algorithm, and LiDAR features shared between vehicles using a V2V ad hoc network. In our proposed framework, each CAV is represented by an actor network that receives compressed LiDAR features along with metadata from nearby CAVs. A schematic representation of the proposed collaborative pipeline is presented in Figure 6.11.

The following is a description of the forward pass of the MARL collaborative mechanism used in a research article:

- Step 1: Preprocessing and feature extraction. Individual LiDAR data is preprocessed and converted into ego vehicle coordinates. The processed data is passed through a series of convolutional layers to create meaningful features.
- Step 2: Compression and V2V communication. The output features are compressed using the DRACO 3D compression algorithm (Google, 2017) to reduce the message size by about 40 times. The agents can share the compressed features in a V2V ad hoc

network. Each CAV then decompresses the message through a decoder, transforms the message to its perspective and aggregates the resulting features into a single tensor.

- Step 3: Action prediction. The MAPPO actor network pipeline is used to determine the action. The input to the MAPPO actor network includes the aggregated LiDAR features.

The algorithm pseudocode is presented in 2.

Algorithm 2 Collaborative MAPPO using LiDAR features.

Initialise θ, the parameters for the critic V and the parameters ϕ for the actor π.
for $ep = 1$ to M **do**
  Retrieve initial state $s_0, O_0, \hat{z}_i$
  for $t = 1$ to T **do**
    for j in *vehicles* **do**
      $h_i^{(0)} = CNN(\hat{z}_i^{(0)})$
      $Compress(h_i^{(0)})$
      $h_s = Aggregate(h_i)$
      Choose action $a_j^t = \pi(O_j^t, h_s; \theta)$
      Calculate value $v^{(j)} = V(s_j^t, h_s; \phi)$
    end for
    Observe $r^t, s^{t+1}, O^{t+1}, \hat{z}_i$.
    Store $[s^t, \hat{z}_i, O^t, a^t, r^t, s^{t+1}, o^{t+1}]$.
    Compute advantage estimate $\hat{A}$ via GAE
    Compute reward-to-go $\hat{R}$ on the stored transitions.
  end for
  Update θ using Adam optimiser on mini-batch
  Update ϕ using Adam optimiser on mini-batch
end for

6.3.2 Experiments

We design two different occluded intersection scenarios that are challenging for vehicles to navigate safely without causing any collisions. These scenarios consist of two CAVs and traffic vehicles. The latter is not part of the V2V network and, therefore, cannot share information (LiDAR data, position, velocity) with nearby CAVs. A Bird's Eye View of the scenarios is presented in figure 6.12. Scenario 1 features an intersection occluded by tall trees, while scenario 2 features a blind summit. Both scenarios are dangerous and consistently lead to collisions. The implementation details of the experiments can be found in the Appendix.



Figure 6.12: Occluded scenarios. The first scenario corresponds to a blind intersection, where vehicles are occluded by tall trees. The second scenario corresponds to a blind summit.

First, we compare our proposed collaborative MARL approach against the following baselines:

- MAPPO with ground-truth locations: CAVs can observe the precise location of each vehicle in the scene, including the occluded vehicle/s. The rest of the pipeline structure is preserved. This baseline should outperform all other methods as it has full observability. The precise location of the vehicles may not be known in practice, or they may be a noisy estimate of the true locations.

- MAPPO with Early Fusion: CAVs share the raw LiDAR point cloud with other nearby agents and use it to predict the action. Sharing the entire LiDAR point cloud with nearby agents to predict actions is impractical due to bandwidth limitations in current communication protocols, such as DSRC and C-V2X (Cui et al., 2022).

- Independent RL: Each ego vehicle uses its own LiDAR observation and metadata to generate the control policy. Agents do not share information with each other and try to maximise an independent reward function. The MAPPO structure is the same for the multi-agent case, except for the message compression and sharing phase.

- Car following model: corresponds to a TTC-based model that calculates the safety time distance to the front vehicle and slows down in case it reaches a threshold. Otherwise, it tries to match the front vehicle's speed.

To facilitate a comprehensive comparison between the proposed approach and the baselines, we employ a set of commonly used metrics, consistent with those utilised in similar studies (Kamran et al., 2020; Isele et al., 2018):

- Average Speed: Average speed per agent across all testing episodes. This is a measure of the overall traffic efficiency.
- Collision Rate: percentage of testing episodes that ended with a collision. This is an indicator of how safe the policies are generated by the different methods.
- Success Rate: The percentage of the test episodes where the agents reached the destination without collisions. A vehicle can successfully avoid collisions but may not reach the desired goal; therefore, the success rate reflects both safety and efficiency.

Baseline comparison

Table 6.4 presents a comparison of the proposed collaborative approach with different baselines. The results represent the average and standard deviation of 100 independent testing runs. The rule-based method has a high average speed but leads to collisions in almost a quarter of the episodes. The independent RL approach outperforms the rule-based method in terms of collision and success rate, but its average speed is relatively low. In contrast, the proposed collaborative method yields efficient driving policies with low collision rates.

Table 6.4: Comparison between the proposed collaborative method to the different baselines. Results correspond to the average of over 1000 independent test runs.

Scenario	Method	Avg. Reward ↑	Avg. Speed (km/hr) ↑	Collision Rate (%) ↓	Success Rate (%) ↑
Occluded Intersection	Rule-based	47.04 ± 12.43	40.45 ± 5.62	25.24 ± 10.15	62.44 ± 10.23
	Independent RL	45.97 ± 16.00	40.65 ± 15.45	10.43 ± 5.53	76.66 ± 17.96
	MAPPO + Ground Truth	68.23 ± 6.06	48.35 ± 16.04	4.87 ± 6.48	89.12 ± 12.77
	MAPPO + Early Fusion	59.03 ± 5.32	44.34 ± 12.35	5.12 ± 4.44	78.45 ± 9.45
	Collaborative MAPPO (ours)	63.35 ± 8.05	45.62 ± 25.30	2.12 ± 4.48	85.34 ± 15.13
Blind Summit	Rule-based	210.25 ± 12.12	60.14 ± 8.16	17.45 ± 10.14	68.24 ± 18.12
	Independent RL	180.45 ± 44.71	68.42 ± 16.72	12.15 ± 8.86	70.71 ± 17.14
	MAPPO + Ground Truth	270.71 ± 23.25	75.04 ± 14.99	1.91 ± 3.32	96.94 ± 10.14
	MAPPO + Early Fusion	235.77 ± 31.75	71.12 ± 21.22	4.24 ± 2.22	90.90 ± 5.28
	Collaborative MAPPO (ours)	257.24 ± 36.76	71.08 ± 20.98	3.07 ± 3.15	94.14 ± 7.15

The collaborative framework demonstrates comparable performance to the use of ground truth locations of all vehicles in the scene. Although the collision rates are similar, the $MAPPO +$

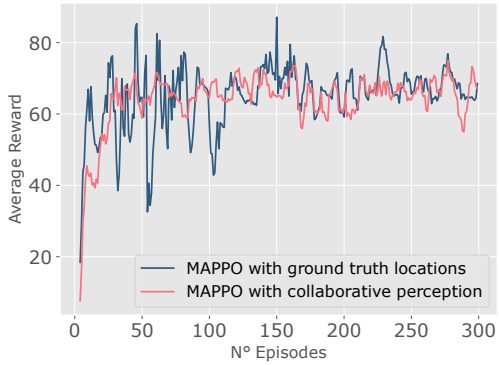


Figure 6.13: Learning curves for MAPPO with ground truth locations of occluded vehicles and MAPPO with collaborative perception.

GroundTruth method achieves higher traffic efficiency and, therefore, higher overall reward. This can be attributed to the fact that the true locations of all vehicles are part of the observation, while the proposed method relies on a fixed range for transmitting and receiving LiDAR features. Thus, agents become aware of occluded vehicles only when other CAVs are within communication range. In real-world situations, having precise vehicle locations may not be feasible, especially when vehicles are occluded by buildings or other obstacles.

Figure 6.13 compares the learning curves of MAPPO trained with ground truth locations of all vehicles and MAPPO with the collaborative perception pipeline. The results, which are an average of 5 independent training runs, show that both methods achieved a similar final average reward. However, the proposed method had a more stable training phase, particularly during the first 100 episodes, which could be attributed to the additional information contained in LiDAR features that aided in stabilising the learning process.

The proposed method effectively meets the bandwidth requirements of current communication protocols such as DSRC and C-V2X. With a frame rate of 20 fps and convolutional features represented as (4, 128, 128), the bandwidth utilisation, after employing the Draco compression algorithm, amounts to approximately 1.075 Mbps. This value remains comfortably below the throughput capabilities of V2X communication protocols, with DSRC at 2.0 Mbps and C-V2X at 7.2 Mbps (Cui et al., 2022). It is worth noting that transmitting the complete point cloud, which consists of 47,240 points, requires approximately 80 Mbps, which exceeds the allowable bandwidth limits.



Figure 6.14: Example trajectory obtained using the proposed method. The CAV coming from the south stops at the intersection even though it has no direct line of sight to the other vehicle.

Figure 6.14 displays an example trajectory generated by the collaborative method. As depicted in 6.14.a), the CAV approaching from the south initiates braking before reaching the intersection, despite having no direct line of sight to the vehicle approaching from the east-west lane, and both vehicles are approaching at a considerable speed. This is possible because of the messages exchanged between the CAVs, allowing the other vehicle to sense the traffic before it enters the intersection. Once the vehicle has turned, the CAV can navigate safely through the intersection towards its destination.

Point Dropout resilience

Real-world liDAR data can suffer from high levels of noise and point dropout (Espadinha et al., 2021). This creates a gap between simulation and reality; thus, control methods should be robust to different noise levels in the sensor reading. To test the robustness of the proposed framework, we simulate different levels of point dropout on the sensor data readings. We tested the trained algorithm with dropout levels ranging from 10% to 40% on the Occluded Intersection scenario.

The proposed method can still perform well in scenarios with 20% LiDAR point dropout or less. Figure 6.15 shows the average test collision rate of the collaborative approach under different levels of point dropout. $20 - 30\%$ dropout represents the threshold region where performance significantly degrades, with collision rates increasing five-fold and, consequently, underperforming compared to independent RL.

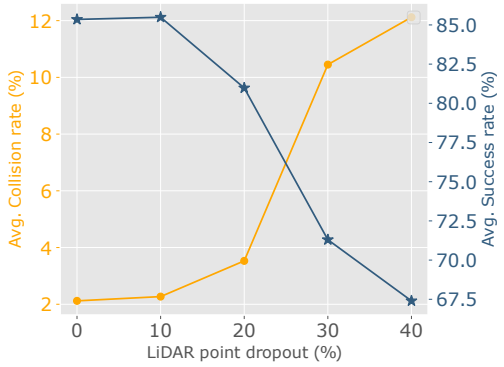


Figure 6.15: Collision and Success rate as a function of LiDAR point dropout.

6.3.3 Summary

Collaborative control methods become essential for occluded scenarios where one or more vehicles are blocked, for instance, by another car or tall building. We presented a decentralised approach based on a local V2V sharing network that can generate cooperative policies in challenging driving settings. The method aggregates LiDAR features of nearby CAVs and uses them to determine the action that maximises traffic efficiency while minimising collision risk. We compared the proposed method against a series of baselines, including a car following baseline, independent RL and a cooperative early fusion method.

Generating collaborative policies leads to lower levels of collisions and higher traffic efficiency. Utilising the proposed method with LiDAR feature sharing, the rate of crashes can be reduced on average by 70% compared to independently trained policies. It also leads to more stable learning, suggesting that using LiDAR features can help stabilise the training procedure. Moreover, 20% of point cloud dropout does not have a great effect on algorithm performance. With higher dropout levels, the performance degrades substantially, leading to underperforming compared to independent RL policies.

6.4 Concluding Remarks

This chapter has illustrated the critical role of strategic communication in enhancing the collaborative capabilities of AVs within complex and dynamic environments. By integrating MARL

with advanced vehicle-to-vehicle communication networks, we have presented two innovative approaches that significantly boost the safety and operational efficiency of AV systems. The first approach focuses on the transmission of vehicle intent through an observation prediction module to forecast and share future policy outputs, effectively reducing collision rates and enhancing coordination. The second method employs a collaborative control strategy utilising compressed LiDAR features for real-time data sharing, proving superior in navigating occluded scenarios where traditional autonomous strategies falter.

Future research in autonomous vehicle communication should address two critical areas to enhance safety and reliability. First, while model-free MARL has shown promise in improving collaboration, its inability to guarantee safety necessitates the exploration of alternative methods that can provide such assurances. Techniques that incorporate constraints or model-based elements may offer potential pathways to achieve more robust safety guarantees. Second, to ensure the practical applicability of these communication strategies, more realistic environments should be explored, noting that MARL-based communication models are computationally intensive, particularly when ensuring coordination across agents. Additionally, a deeper investigation into HDV-CAV dynamics is required, with refined interaction models to better capture mixed traffic scenarios within a MARL framework. Real-world operational factors such as bandwidth usage, latency, packet loss, and network scalability must also be considered to develop resilient communication protocols capable of handling real-world complexities.

Moreover, while enhancing communication improves vehicle collaboration, it also introduces significant vulnerabilities, such as cybersecurity risks and the threat of malicious attacks. These issues underscore the need for secure communication protocols, a topic that will be explored in the next chapter. Ensuring security is crucial for deploying collaborative autonomous vehicle technologies, requiring rigorous protocols to prevent breaches and ensure reliable communication.

Chapter 7

Robust Defence Against V2X Communication Attacks

This chapter is adapted from the following publication:

Parada, L. and Angeloudis, P. (2024). Securing autonomous vehicle networks: Strategic defence against communication attacks. In RSS 2024 Workshop for Autonomous Vehicles Across Scale (AVAS), Delft, Netherlands.

7.1 Introduction

At the tactical level of autonomous decision-making, Multi-Agent Reinforcement Learning (MARL) with communication provides significant benefits for AV systems, particularly in scenarios requiring high levels of coordination. However, as highlighted in the previous chapter, the reliance on Vehicle-to-Everything (V2X) communication introduces new challenges, especially with the increasing prevalence of adversarial threats. These attacks compromise the integrity of V2X communications, posing significant risks to overall traffic safety. This chapter explores strategies to defend against V2X communication threats at the tactical decision-making level (see figure 7.1).

This emerging challenge necessitates a deeper understanding of how deceptive information impacts multi-agent operations and requires the development of robust countermeasures. Unlike traditional adversarial attacks, which focus on manipulating a single vehicle's perception or decision-making (Zhang et al., 2021a; Li et al., 2023a; Pattanaik et al., 2017), communication attacks exploit the interdependence of V2X protocols (Tu et al., 2021), disrupting the coordination and collaboration critical at the tactical level of decision-making.

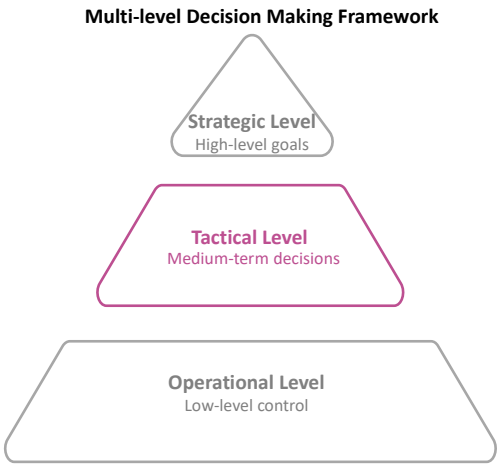


Figure 7.1: Robust defence strategies against V2X communication attacks at the tactical decision-making level.

Given the critical role V2X technology plays in enabling interaction between vehicles and their environment, cybersecurity in V2X communications has become a critical area of research. V2X technology, which includes vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-pedestrian (V2P) communications, is vulnerable to various attacks such as denial of service (DoS), spoofing, and man-in-the-middle attacks. These attacks target Vehicle Ad-Hoc Networks (VANETs), compromising data integrity and vehicle control systems. To mitigate these risks, industry standards like IEEE 802.11p (IEEE, 2010) for Dedicated Short-Range Communications (DSRC), IEEE 1609 (IEEE, 2016) for higher-layer security protocols, and ISO/SAE 21434 for managing automotive cybersecurity risks have been established. A comprehensive review of V2X cybersecurity is provided in (Kim et al., 2021).

Adversarial Deep Learning attacks on V2X communications represent a newer, less-explored research area, mostly addressed through imitation learning. Research has shown that even minor perturbations can result in potent attacks, and adversarial training has been found effective when the attack model is known (Tu et al., 2021). To defend V2V attackers, some methods rely on sampling a subset of messages and verifying consensus between object detections (Li et al., 2023b). Recent studies have demonstrated that introducing perturbations into LiDAR-based V2X systems can enhance performance by training on such altered scenes (Xiang et al.,

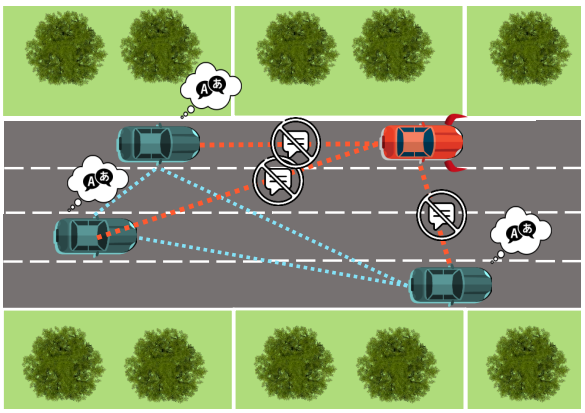


Figure 7.2: Conceptual overview of the proposed defence mechanism to maintain network integrity by detecting adversaries and reconstructing their messages.

2023). However, these approaches typically pre-generate adversarial images or point clouds for training, thus lacking adaptability. By framing the issue as a communicative MARL problem, it becomes possible to train adaptive attackers, resulting in more robust defence mechanisms.

Despite the limited research on adversarial communication in MARL, key studies have begun to shed light on the complexity of defence mechanisms and adversarial strategies. For instance, Blumenkamp and Prorok (2020) observed that agents can develop adversarial communication strategies driven by competitive incentives rather than adversarial design. Xue et al. (2022) developed deliberative adversarial communication methods, while Mitchell et al. (2020) and Tu et al. (2021) explored message detection methods and adversarial attacks on feature maps, highlighting significant vulnerabilities. Despite advancements in understanding adversarial communication and detection, practical defence strategies are limited. Our research aims to bridge this gap by introducing a modular defence architecture that extends beyond traditional detection and prediction to mitigate adversarial threats.

This chapter presents TrustCommNet, an end-to-end learnable architecture that enhances autonomous driving security by ensuring communication integrity in autonomous vehicle networks. TrustCommNet proactively defends against adversarial threats through real-time checks and adaptive responses. A conceptual overview is presented in figure 7.2. Its robustness comes from its key components: a system to validate communication integrity and filter malicious content, a message scheduling system for effective timing and targeting, and a message reconstruction process to correct distorted messages.

To test our proposed defence mechanism, we craft an attacker model based on Projected Gradient Descent (PGD) (Kurakin et al., 2017) to execute ℓ_∞-bounded perturbations on communication messages. This mechanism effectively challenges the communication processes within MARL systems, as we will show that it disrupts standard decision-making and policy trajectories. We use this model for two purposes. Firstly, we rigorously stress test various state-of-the-art communication MARL algorithms (Niu et al., 2021; Das et al., 2020; Liu et al., 2019; Singh et al., 2018) with PGD to demonstrate how even minor perturbations in communication can significantly impact team performance. Secondly, we use PGD as a rigorous benchmark to evaluate the resilience of our proposed defence framework. This framework is tested in both classical MARL environments and the Highway-env environment (Leurent, 2018). The results demonstrate our framework's ability to detect and counteract sophisticated attacks effectively, quickly recovering from disruptions and restoring previous levels of cooperative performance. Through various experiments, we show that our framework achieves near-perfect zero-shot performance in challenging environments and can scale to accommodate multiple attackers.

7.2 Problem Formulation

7.2.1 Communication-Based MARL

We explore communication-based MARL as an extension of a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). This framework is formally represented by the tuple $\langle N, S, A, O, \mathcal{T}, R, \gamma, \mathcal{M}, \mathcal{C} \rangle$. Within this framework, N denotes a set of agents $N = \{1, 2, \ldots, n\}$, S represents the global state space, and A encompasses the joint action space, with each agent i having individual actions $a_i \in A$ and observations $o_i \in O$. The system's dynamics are governed by a transition function $\mathcal{T} : S \times A \to S$, which determines the next state $s' \in S$ given the current state $s \in S$ and joint action $a = (a_1, \ldots, a_N) \in A$. The reward function $R : S \times A \times S \to \mathbb{R}$ assigns a scalar reward to the transition from state s to s' due to the joint action a.

A key feature of communication-based MARL is the integration of communication between agents, captured by the message set $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$ and a communication protocol $\mathcal{C} : \mathcal{M}^N \to \mathcal{M}$, which consolidates messages from all agents into a single message. Each agent i operates with two policies: the action policy $\pi_i(a_i|o_i, m_{rec,i})$, determining the action a_i based on the observation o_i and received message $m_{rec,i}$; and the message policy $\xi_i(m_i|o_i)$, generating the message m_i to be shared with other agents based on o_i. We represent messages as continuous vectors of size n_{emb}.

The training goal is to maximise the expected cumulative discounted reward, as defined by the objective function $\max_{\pi_1,\pi_2,\ldots,\pi_N} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r^t(s^t, a_1^t, a_2^t, \ldots, a_N^t)\right]$, where γ is the discount factor and T the time horizon.

7.2.2 Adversarial MARL

Building on the Byzantine Generals problem (Lamport et al., 2019), which explores consensus difficulties in the presence of unreliable agents, the Adversarial MARL approach addresses the uncertainty related to the number and strategies of adversarial agents. We consider $N_{adv} \subseteq N$ adversarial agents equipped with the same observational capabilities as standard agents but are also capable of crafting malicious messages m_{adv} to mislead others, thereby disrupting decision-making and compromising collective goals. Their behaviour is guided by an adversarial message policy $\xi_{adv}(m_{adv}|\tau, m)$, where τ is the history of observations and actions and m is the unaltered message produced by the agent through their message policy. Thus, the adversarial agents aim to introduce carefully crafted message alterations to damage the overall team performance.

In the domain of adversarial robustness, the concept of the saddle point problem emerges as an important theoretical underpinning. This can be described as a minimax optimisation problem, which provides a unified formulation for both adversarial and defence strategies (Madry et al., 2019). In the case of imitation learning, this problem is defined by the following equation:

$$\min_{\theta} \max_{\epsilon \in \mathcal{E}} L(\theta, x + \epsilon, y) \tag{7.1}$$

where θ represents the defence model parameters, x and y are the input data and its corresponding label, ϵ is the adversarial perturbation corresponding to an allowable set $\mathcal{E}$, and L is the loss function. This problem seeks to find the model parameters θ that minimise the maximum loss induced by any adversarial perturbation within the set $\mathcal{E}$.

To address the dynamics between non-adversarial and adversarial agents, we formulate the Adversarial MARL problem as an optimisation task focused on maximising expected returns for cooperative agents while minimising them for adversarial agents:

$$\max_{\theta} \min_{\epsilon} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r^t \left(s^t, \mathbf{a}^t, m^t_{adv}(\epsilon)\right)\right] \tag{7.2}$$

7.3 Framework

This section outlines our methodology for countering adversarial attacks on communication in cooperative MARL. Our framework includes an adversarial algorithm using ℓ_∞-bounded perturbations employing Projected Gradient Descent to reduce the aggregated reward. For defence, we employ TrustCommNet, a trust-based validation network to ensure message integrity, supported by a Theory of Mind (ToM) predictor. These components will be explained in the following subsections. The framework's architecture is illustrated in Figure 7.3.

7.3.1 Dynamic ℓ_∞-bounded Attack on Communication

Designing a strategic DRL attacker mainly involves addressing two challenges: determining "how" and "when" to launch attacks. Our "how-to-attack" strategy introduces subtle message perturbations to create a disparity between the network's predicted and actual outputs, systematically deviating agents' policies from their optimal paths. Utilising the differentiability inherent of the system, this method efficiently computes necessary gradients without needing full environmental knowledge.

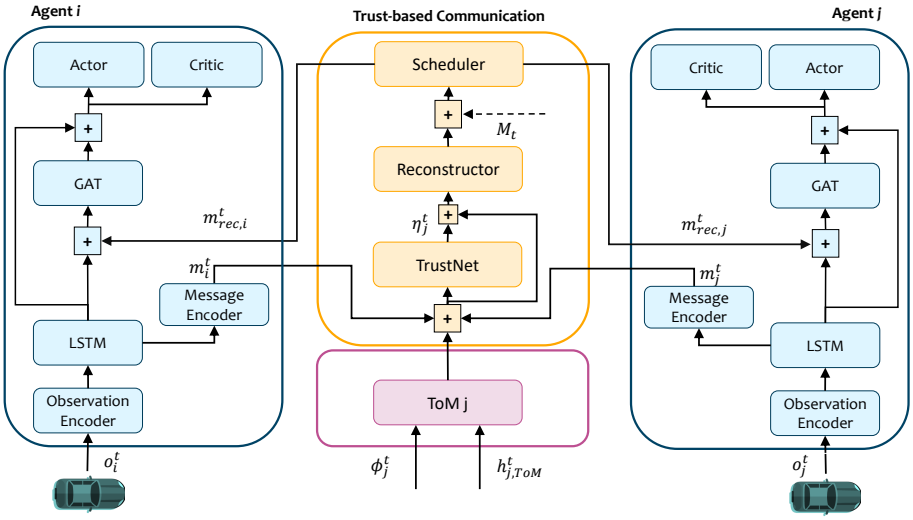


Figure 7.3: This schematic presents TrustCommNet, our proposed MARL defence architecture, highlighting its key components: a Theory of Mind predictor for agent interactions, a message validation and reconstruction mechanism, and a scheduler for efficient bandwidth use.

We use PGD to optimise the adversarial message m_{adv} and target the team loss within the MARL context. Employing PGD enables precise manipulation of the agent's messages to maximise the policy loss while operating within the ℓ_∞-bound, thus striking a crucial balance between the efficacy of the attack and undetectability. The PGD update for the adversarial message is articulated as follows:

$$m^{(t+1)} = Proj_{m+\epsilon}\left(m^t + \alpha \cdot sgn\left(\nabla_{m^t} L(\theta, m^t, h^t)\right)\right) \tag{7.3}$$

where $m^{(t+1)}$ is the updated adversarial message, with sgn denoting the sign function, and h^t as the current hidden state. The step size is given by α, and $Proj_{m+\epsilon}$ maintains the message within the allowable perturbation bounds, ϵ. We utilise a cross-entropy loss function that maximises the probability of the worst possible actions. This approach guides adversarial agents to craft messages that effectively degrade the performance of non-adversarial agents by increasing the likelihood of sub-optimal decisions.

We now shift focus to the "when-to-attack" problem, which introduces an additional layer of complexity to the task of identifying potential attackers within a system. Our methodology for timing attacks hinges on a specifically designed preference function for the attacker. We adopt a policy output disparity function, similar to the one proposed by Lin et al. (2019). This

function evaluates the disparity between the maximum and minimum outputs of an agent's policy at a given step:

$$c = \max \pi_i(o_i^t, m_{rec,i}^t) - \min \pi_i(o_i^t, m_{rec,i}^t) \tag{7.4}$$

The rationale behind employing this function is to evaluate the decisiveness of a DRL agent towards a particular action; a larger difference indicates a strong preference for a specific action, justifying an attack. Conversely, a minimal difference suggests an absence of clear preference, advising against an attack on this time step. To operationalise this strategy, we introduce a threshold β, serving as the critical value for $c(o_i^t)$ above which an attack is deemed worthwhile.

We acknowledge there are more complex attacking strategies in the literature, such as state luring mechanisms (Lin et al., 2019; Mo et al., 2023); however, our focus in this study is on the defence mechanism. Thus, we just want to create a simple yet effective attack on multi-agent communication.

7.3.2 Trust Verification defence Mechanism

Building on the concept of Machine Theory of Mind (ToM) (Rabinowitz et al., 2018), our model enables agents to predict the communication messages from others. While similar to the ToM2C approach (Wang et al., 2022), which infers goals and observations from pose and encoded observation, our method predicts encoded communication messages of other agents based solely on their pose ϕ_j.

First, as illustrated in figure 7.3, the imagined message for an external agent j from the point of view of an agent i is $f_{ToM,j,i}(m_{j,i,ToM}^t | \phi_j^t, h_{j,i,ToM}^t)$. This model considers as input agent's j pose ϕ_j^t, and a hidden state $h_{j,i,ToM}^t$ that represents the history of past behaviour. From this point onwards, we drop the index of agent i for simplicity. It should be noted that figure 7.3 only displays the modelling network of agent j from the perspective of agent i, $f_{ToM,j,i}$. However, each agent has an independent network to model the communications of others. Since all agents'

communication messages have the same form, they can share the weights of the network $f_{ToM,j}$. In practice, we use an LSTM followed by a fully connected layer to represent $f_{ToM,j}$.

Second, we introduce and train a network layer, which we name TrustNet, designed to assess the reliability of each agent's communication. TrustNet operates by comparing the predicted communication message with the actual message sent by each agent. Specifically, for each agent j at time t, TrustNet evaluates $f_{Trust}(\eta_j^t | m_j^t, m_{j,ToM}^t)$, where η_j^t acts as a probabilistic trust indicator. This indicator approximates the agent's reliability by quantifying the discrepancy between its expected message $m_{j,ToM}^t$ and the actual communication message m_j^t. We assume the homogeneity of all agents' observations and messages. Thus, we assign the same network parameters for f_{ToM} and f_{Trust} across agents.

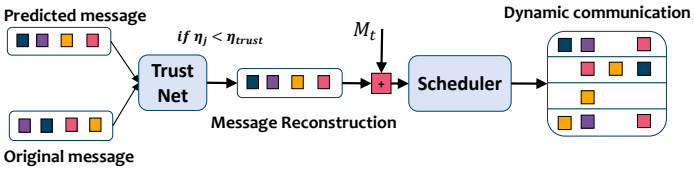


Figure 7.4: Trust-based mechanism for detecting, reconstructing and scheduling messages in the communication network.

To safeguard the integrity of multi-agent communications, our framework implements a dynamic trust-based protocol. In instances where an agent's trust score descends below a trust threshold $\eta_j^t < \eta_{trust}$, the system activates protective protocols and the agent's communication message will then be blocked and a new message can be reconstructed based on the predicted message $m_{j,\text{ToM}}^t$ and agent's i current observation o_i:

$$m_{j,\text{new}}^t = \begin{cases} f_{Rec}(m_{j,\text{ToM}}^t, o_i) & \text{if } \eta_j^t < \eta_{\text{trust}} \\ m_j^t & otherwise, \end{cases} \tag{7.5}$$

Following message reconstruction, a communication scheduler processes the updated messages from all agents, generating an adjacency matrix G^t that defines the communication scheme

using a hard attention mechanism. To maintain differentiability during training, a Gumbel-Softmax layer was employed. The matrix G^t indicates active communication with $G^t_{ij} = 1$ for agent i to agent j at time t and $G^t_{ij} = 0$ otherwise. A simplified schematic overview of the trust-based message detection, reconstruction, and scheduling process is shown in figure 7.4.

Agents then receive incoming aggregated messages $m^t_{rec,i}$ and integrate them with their own encoded observation using a Graph Attention Network (GAT). This is then utilised to predict actor and critic outputs as depicted in figure 7.3. This structured approach allows agents to dynamically refine their communication strategies, enhancing message efficiency and the robustness of the network against potential threats.

7.3.3 Overall Training Objective

To enable agents to develop accurate predictions of other agents' communications, we use the Mean Squared Error (MSE) between the predicted and actual communication values:

$$\mathcal{L}_{ToM} = \frac{1}{N} \sum_{j=1}^{N} \sum_{i \neq j}^{N-1} \left(m^t_j - m^t_{j,ToM} \right)^2 \tag{7.6}$$

where N is the number of agents, m^t_j is the actual communication value for the j-th message sample, and $m^t_{j,ToM}$ is the corresponding predicted value by the ToM layer. The MSE-based loss is crucial for accurately measuring the differences between the model's predicted values and actual communication behaviours, ensuring effective learning of communication dynamics in the multi-agent environment.

Additionally, the training process employs an Actor-Critic loss ($\mathcal{L}_{AC}$) to optimise the agents' policies towards maximising expected discounted rewards. This loss integrates a policy gradient component, optimising the agents' policy networks, and a value component, refining the value function estimates:

$$\mathcal{L}_{AC} = \mathbb{E}_{(\tau, m_{adv})} \left[\sum_{t=0}^{T} A^t \cdot \nabla_\theta \log \pi_i(a_i^t | o_i^t, m_i^t) \right] + \lambda \sum_{t=0}^{T} \nabla_\rho (R_i^t - V_\rho(o_i^t))^2 \tag{7.7}$$

where $A^t = (R_i^t - V_\rho(o_i^t))$ is the advantage function and V_ρ is the value function with set of parameters ρ. In practice, we let the policy and the value function share the same parameters of the neural network except for their respective output layers (heads). The parameter λ balances the two components, facilitating effective policy learning and decision-making. Therefore, the overall training loss of the model is $L_{tot} = L_{ToM} + L_{AC}$.

We use the Centralised Training Decentralised Execution (CTDE) paradigm. During training, agents utilise observations and communication from others, but during inference, they rely on local observations and the scheduler-determined adjacency matrix G^t. This framework assumes agents know each other's locations to predict communication patterns without accessing others' observations or messages.

7.4 Experiments

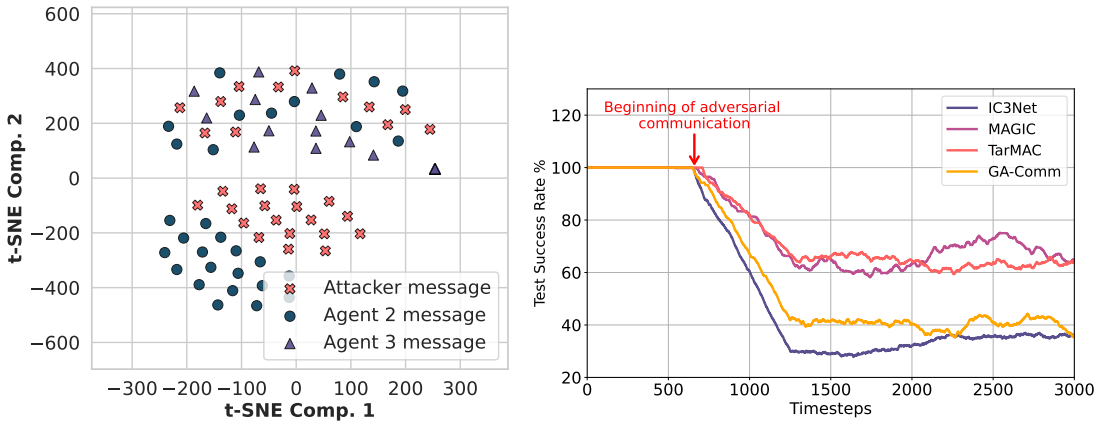
In this section, we present the experimental results from testing the adversarial defence capabilities of various state-of-the-art communicative MARL algorithms, including our proposed TrustCommNet. The baseline algorithms evaluated include MAGIC (Niu et al., 2021), IC3Net (Singh et al., 2018), TarMAC (Das et al., 2020), and Ga-Comm (Liu et al., 2019). We first benchmark these algorithms in two classical multi-agent environments: Predator-Prey and Traffic Junction. Following this, we test the algorithms on autonomous driving systems using the modified version of the Highway-env environment (Leurent, 2018) presented in Chapter 5.

We initially trained the algorithms without adversarial messages to establish a baseline for comparison. We exposed these algorithms to adversarial messages using our PGD-based attack to assess their immediate defence performance. Adversarial training was then used to evaluate the extent and speed of recovery.

Our primary evaluation metric is the average success rate (%), or win rate, across all environ-

ments and scenarios, indicating the proportion of episodes successfully completed. Additionally, we assess performance using the average team return. To ensure the robustness of training results, we employ 5 independent seeds to generate learning curves, and we validate our experiments across 100 independent test episodes. Additional details of the training procedure and the implementation details are shown in Appendix A.

7.4.1 Assessing Communication Vulnerabilities in MARL



(a) (b)

Figure 7.5: Communication attacks on MARL. (a) t-SNE visualisation shows adversarial messages can mimic legitimate communications. (b) Adversarial communication leads to a marked decline in success rates for all MARL baselines.

We first demonstrate the complexity of identifying adversarial messages in a multi-agent framework through a 2D t-SNE visualisation in Figure 7.5a, compressing the 128-sized encoded message vectors for graphical representation, using a value of $\epsilon = 0.3$ used for this analysis. Despite distinct patterns in legitimate interactions, adversarial messages blend with benign communications, emphasising the need for advanced detection mechanisms.

Figure 7.5b and Table 7.1 analyse the vulnerability of MARL algorithms to adversarial attacks. Introducing a PGD-based attacker at the 500-timestep mark causes a sharp decline in test success rates, especially for IC3Net and GA-Comm, which fall below 40%. The table shows high variability post-attack, particularly at higher perturbation levels ($\epsilon = 0.75$ and $\epsilon = 1.0$), indicating unpredictability. While all methods are affected, TarMAC and MAGIC exhibit less deviation and smaller success rate drops, suggesting a certain level of robustness.

Table 7.1: Performance degradation of MARL algorithms under different levels of adversarial communication.

Method	$\epsilon = 0.5$	$\epsilon = 0.75$	$\epsilon = 1.0$
TarMAC	68.2 ± 10.4	60.4 ± 4.0	64.7 ± 7.4
IC3Net	72.9 ± 7.3	53.7 ± 11.3	38.0 ± 12.4
GA-Comm	82.7 ± 5.6	49.1 ± 5.4	36.2 ± 6.5
MAGIC	67.8 ± 9.2	62.0 ± 3.9	65.4 ± 6.0

7.4.2 Efficacy of the defence Mechanism

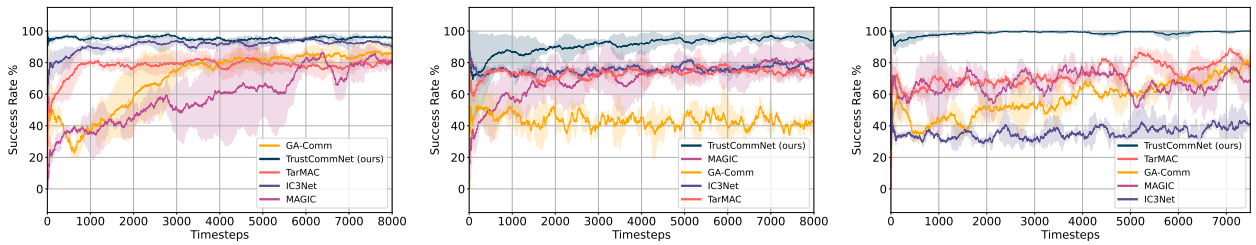
Traffic Junction and Predator Prey

This subsection presents the robustness results of TrustCommNet and baseline models against communication attacks in two classical environments: Predator-Prey and Traffic Junction. The adversarial training procedure, as well as the environment description, are presented in Appendix A.

Theoretically, retraining baseline algorithms should provide a defence against malicious agents for two reasons. First, agents may learn to disregard or mitigate adversarial communication. Second, these algorithms are not exclusively dependent on communication for policy and value predictions. Instead, they integrate a combination of encoded messages with their own observations and hidden states. Thus, they can adapt the weights of the network to prioritise observations, although this may reduce cooperation effectiveness.

Figure 7.6 illustrates the efficacy of our method during adversarial training compared to baseline approaches. Results correspond to the average and standard deviation over the five different seeds. Notably, TrustCommNet demonstrates a swift and robust recovery, approaching full performance levels promptly while minimising disruptions caused by the malicious agent. In contrast, the adversarial training curves for the baseline methods show a lag in recovery, with most of them failing to regain their initial success rates. Additional results on the team reward are presented in Appendix A.

Table 7.2 offers a comparative analysis of the various methods. Key observations from Figure 7.6 and Table 7.2 reveal that IC3Net recovers quickly with minimal disruption in simpler envi-



(a) TJ - 5 agents (b) TJ - 10 agents (c) PP - 5 agents

Figure 7.6: Adversarial training performance comparison, showing near-perfect zero-shot performance and rapid recovery of TrustCommNet against various baselines over time.

ronments like TJ-5, yet struggles with efficiency in more complex settings like PP-5, showing inconsistent learning patterns. Conversely, TarMAC exhibits stable recovery across all scenarios, with a smaller initial performance drop and a resilience attributable to its architecture, where communication inputs influence only the policy head, not the value head.

Table 7.2: Test performance after adversarial training. TJ=Traffic Junction, PP=Predator-Prey.

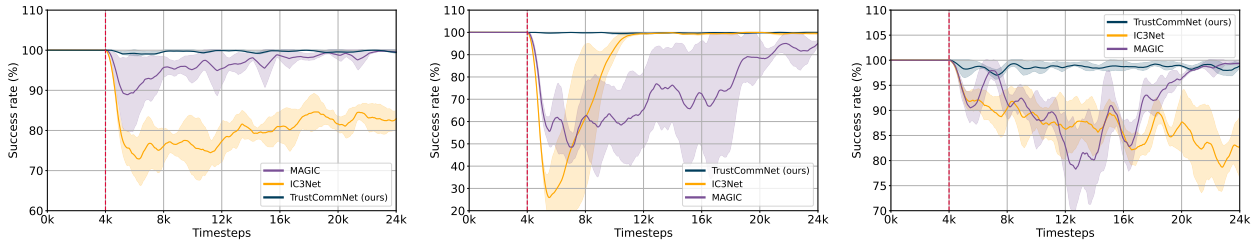
Method	TJ - Easy	TJ - Med	PP - Med
TarMAC	73.8 ± 6.0	77.7 ± 4.4	60.2 ± 9.4
IC3Net	90.8 ± 3.1	81.2 ± 2.6	34.5 ± 7.2
GA-Comm	93.1 ± 3.4	39.4 ± 13.4	68.2 ± 6.2
MAGIC	86.3 ± 5.6	83.0 ± 3.6	72.5 ± 7.7
TrustComm	$\mathbf{95.6 \pm 2.2}$	$\mathbf{97.8 \pm 2.9}$	$\mathbf{99.8 \pm 0.2}$

Highway-env

Once validated on the classical MARL environments, we used our modified version of Highway-env, presented in the previous chapter, to assess the performance of the defence mechanism. This environment is more complex, involving a sophisticated observation space (LiDAR map) and highly dynamic situations such as lane merging and busy unsignalised intersections. For additional details on the environment, please refer to the previous chapter. For these experiments, we only used MAGIC and IC3Net as baselines since they were the only ones able to achieve perfect or near-perfect performance in fewer than 1×10^7 steps.

Figure 7.7 presents the result comparison of performing adversarial training for 2×10^5 timesteps. Moreover, the zero-shot performance over 100 test episodes of the different algorithms is pre-

sented in table 7.3. Our method requires minimal training and, in some cases (intersection and highway), demonstrates zero-shot safety against adversarial threats without previous exposure to the communication attack. In contrast, both MAGIC and IC3Net are severely affected by adversarial communication, displaying high levels of disruption (low success rate) and long recovery tails. IC3Net is not able to recover in two out of three scenarios in 2×10^5, while MAGIC is able to recover performance in all three scenarios but takes several timesteps to reach previous performance levels.



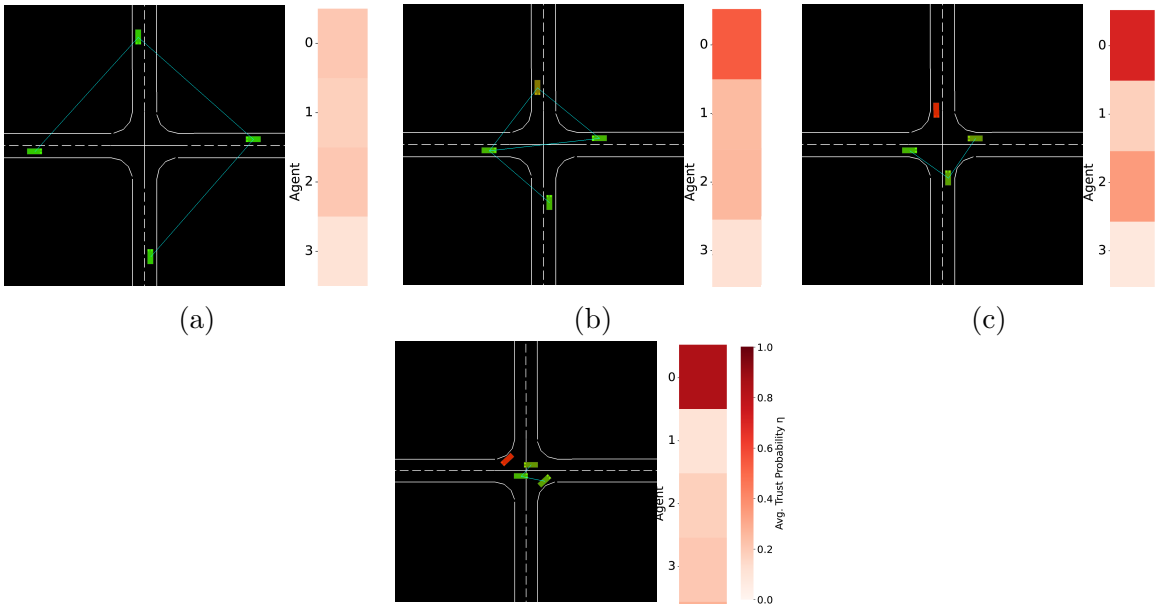
(a) Highway (b) Intersection (c) Lane Merging

Figure 7.7: Performance comparison of adversarial training between TrustCommNet and baseline models. Adversarial communication and training commenced at the 4000-step mark, indicated by the dashed red line.

Table 7.3: Zero-shot performance of different algorithms for the highway-env environment.

Method	Highway	Intersection	Lane-Merging
IC3Net	75.5 ± 8.5	34.0 ± 12.45	32.5 ± 9.5
MAGIC	90.83 ± 6.4	52.04 ± 25.3	84.0 ± 12.8
TrustCommNet	99.3 ± 1.2	99.0 ± 1.3	97.6 ± 2.1

Figure 7.8 shows TrustCommNet in action in the Intersection scenario with a single adversarial agent (agent 0). The figure shows the communication graph and normalised average trust probability for each agent (heatmap on the side). Initially, all agents share messages to locate the prey (frames (a) and (b)). As the episode progresses, agent 0's malicious behaviour is detected, leading to an increase in its average trust score (heatmap) and subsequent blocking by other agents (frames (c) and (d)). By reconstructing the original message, agents can navigate the intersection safely.



(a) (b) (c)

(d)

Figure 7.8: Sequence of an episode of the Intersection scenario featuring a single attacker agent (agent 0). The figures illustrate part of the defence strategy, which involves blocking the communication of the corrupted agent.

7.4.3 Different Levels of Attack Intensity

Figure 7.9 presents the success rates of TrustCommNet and the baseline algorithms under varying attack intensities, denoted by ϵ^t. The attack intensity was varied from 0.1 to 5 in increments of 0.1, revealing distinct performance outcomes. It is important to note that none of the algorithms were trained with exposure to these attacks. The figure clearly demonstrates that TrustCommNet maintains substantial robustness against high attack intensities, with its success rate never dropping below 80%, even as ϵ^t increases. In contrast, IC3Net and MAGIC exhibit significant declines in success rates, even at relatively low ϵ_t values (less than 1.0).

7.4.4 Scaling to Multiple Attackers

We evaluated the performance of our proposed defence mechanism against varying numbers of attackers in the Intersection scenario with up to five agents, using the zero-shot success rate (%) as the metric. The results, presented in Table 7.4, show the rapid performance decline of the IC3Net benchmark as the number of attackers increases, while MAGIC exhibits stable

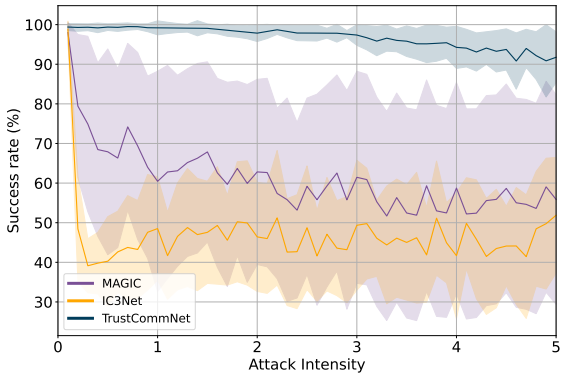


Figure 7.9: Performance of TrustCommNet and Baselines with varying attack intensity.

Table 7.4: Scalability of zero-shot algorithm defence across different numbers of attackers.

N° Attackers	MAGIC	IC3Net	TrustCommNet
1	41.3 ± 6.6	25.6 ± 12.45	99.2 ± 1.4
2	44.5 ± 24.7	26.0 ± 6.7	99.0 ± 1.3
3	41.7 ± 20.9	19.6 ± 5.5	99.0 ± 1.3
4	44.1 ± 26.7	16.2 ± 4.2	98.8 ± 1.3
5	45.1 ± 28.29	3.8 ± 3.3	99.3 ± 1.2

but poor performance. In contrast, TrustCommNet maintains robust performance across all attacker counts, demonstrating its potential to scale effectively in larger networks with multiple attackers. These findings are also depicted graphically in Figure 7.10.

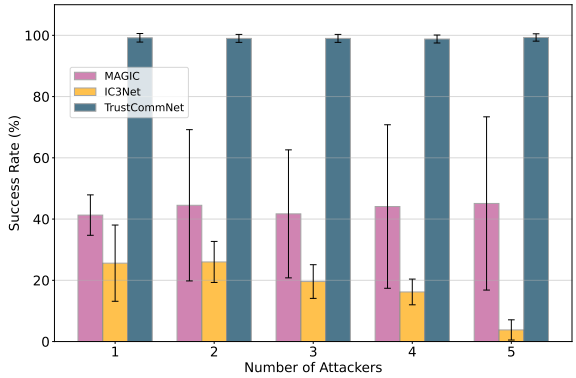


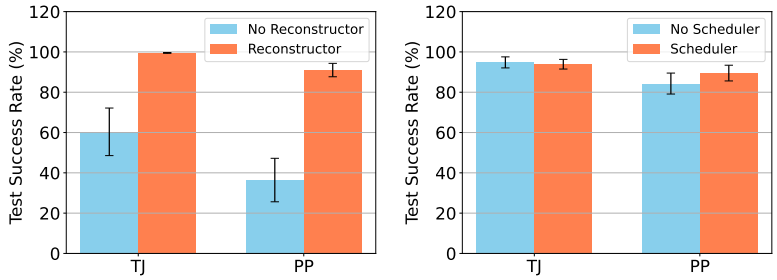
Figure 7.10: Scalability of TrustCommNet and Baselines with varying numbers of attackers.

7.4.5 Ablation studies

We evaluated the contribution of the proposed method by removing two key components of the architecture: the Message Reconstructor and the Scheduler. We test the effect of using

the Message Reconstructor in the classical environments and show the results in figure 7.11. The ablation study on the Message Reconstructor module shows a significant performance drop without it, suggesting a critical role.

Results of the ablation study of the Communication Scheduler can be seen in table 7.5. The ablation study reveals that incorporating a communication scheduler significantly lowers graph density across all methods, indicating improved communication efficiency. However, its impact on success rates varies, with a minor reduction observed in PP and a marginal difference in TJ, highlighting the scheduler's role in balancing communication overhead against operational effectiveness.



(a) Message Reconstructor. (b) Message Scheduler.

Figure 7.11: Results of the ablation studies.

Table 7.5: Ablation study: Impact of the Communication Scheduler on graph density and success rates.

	Method	Graph Density	% Success
PP	Scheduler	0.47 ± 0.07	89.5 ± 3.9
	No Scheduler	1.0 ± 0.0	84.3 ± 5.2
TJ	Scheduler	0.76 ± 0.1	93.9 ± 2.4
	No Scheduler	1.0 ± 0.0	94.8 ± 2.74

7.5 Concluding Remarks

This chapter explored the vulnerabilities of MARL and Vehicle-to-Everything (V2X) communication to adversarial attacks and provided a modular defence architecture to counter these threats. Communication-based MARL offers significant benefits for autonomous driving systems, particularly in scenarios requiring high levels of coordination. However, the increasing

prevalence of adversarial threats in V2X communications necessitates new solutions to ensure secure and efficient collaboration. This emerging challenge requires a deeper understanding of how deceptive information impacts multi-agent operations and the development of effective countermeasures.

Our modular defence framework was designed to enhance the robustness of autonomous driving systems against adversarial communication. We model the adversarial communication problem as a MARL problem and devise a defence mechanism that is able to counteract the attacks. Our comprehensive evaluation against ℓ_∞-bounded adversarial attack demonstrates how message alterations can significantly diminish team performance in state-of-the-art MARL algorithms. Although baseline frameworks provide certain defences by adversarial training, this process often disrupts network cooperation, leading to lower success rates. In contrast, TrustCommNet shows near-perfect zero-shot performance in all driving scenarios, surpassing retraining conventional communication MARL algorithms.

Future research should focus on enhancing scalability to support densely populated environments and complex communication networks. Additionally, exploring different forms of attack, including point cloud dropout, spoofing, and signal denial, will further strengthen defence mechanisms. Another critical area is developing adaptive defence algorithms that can dynamically adjust to new types of threats and evolving attack strategies, ensuring continuous protection and resilience in autonomous driving systems.

Chapter 8

Scaling Deep Reinforcement Learning For Autonomous Fleet Management

Parts of this chapter have been adapted from the following publication:
Ngu, E., Parada, L., Escribano Macias, J. J., & Angeloudis, P. (2022, November). Decentralised Multi-Agent Reinforcement Learning Approach for the Same-Day Delivery Problem. Transportation Research Record: Journal of the Transportation Research Board, 2676 (11), 385–395.

8.1 Introduction

While the previous chapters have focused on tactical and operational decision-making for Autonomous Vehicles (AVs), this chapter shifts to the strategic-level challenge of managing autonomous fleets for mobility services—a critical component for the mass deployment of AVs. Effective fleet management is crucial for scaling AV systems since mass mobility services have the potential to reduce traffic congestion and increase transportation efficiency. At the strategic level, objectives such as minimising passenger wait times, optimising vehicle utilisation, and managing fluctuating demand are vital for ensuring the success and scalability of autonomous mobility networks.

Autonomous fleet management involves making decisions on dynamic dispatching, routing, and rebalancing to meet demand across a wide geographic area. Factors such as passenger wait times, vehicle utilisation, and operational costs are critical, requiring algorithms that can manage thousands of vehicles in a coordinated and scalable manner. While tactical coordination focuses on vehicle interactions, the strategic level emphasises optimising overall fleet performance to meet broader operational goals. This chapter explores how Multi-Agent Reinforcement Learning (MARL) can be scaled to address these challenges, ensuring autonomous vehicles collaborate effectively to improve efficiency and reduce passenger waiting times.

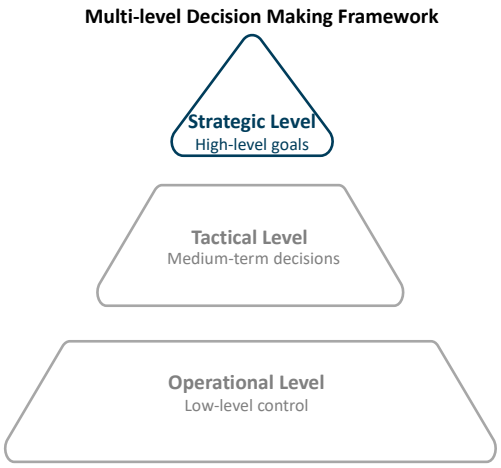


Figure 8.1: Autonomous fleet management at the strategic decision level.

The need for efficient autonomous fleet management methods has grown as consumer mobility patterns have transformed and continue to evolve. Mobility-on-demand (MoD) services, like those offered by Uber and Didi Chuxing, have gained widespread popularity. Technological advancements in autonomous vehicle technology will soon enable dedicated, personalised transportation services, known as Autonomous Mobility on Demand (AMoD) (Simoudis, 2020; Pavone, 2015). Companies such as Waymo and Cruise are currently trialling AMoD services, aiming for larger-scale implementation soon. These trials mark a significant advance in autonomous transportation, potentially revolutionising the industry (McGee, 2021). However, several critical challenges for these mobility systems remain.

One of the biggest challenges is the efficient redistribution of empty vehicles to areas with high demand and low supply, commonly known as the fleet rebalancing problem. While this issue is also present in traditional Mobility on Demand (MoD) services, it presents a unique challenge for AMoD due to the ability of autonomous vehicles to self-rebalance to different areas as demand patterns change. AMoD services require online or real-time solutions for Dynamic Vehicle Routing Problems (DVRPs), which are known to be computationally difficult to solve

(Psaraftis et al., 2016). Exact optimisation solutions for DVRPs are rarely feasible beyond a few hundred vehicles and customers (Bertsimas et al., 2019). As such, developing effective solutions for the fleet rebalancing problem is a critical factor in the successful deployment and operation of AMoD services.

Currently, fleet operators make most rebalancing decisions centrally, often employing techniques like surge pricing to manage demand and supply dynamically. While centralised methods aim to optimise the system's global objective, they face scalability issues. For instance, a branch-and-cut algorithm for four vehicles and 48 requests can take up to four hours (Cordeau, 2006). For MoD services, which involve thousands of vehicles and hundreds of thousands of requests per hour, real-time operation is essential. Heuristic algorithms offer a natural alternative, providing near-optimal solutions in an acceptable time, but they require hand-crafted features and expert knowledge (Horn, 2002; Shuo Ma et al., 2013; Peng et al., 2020).

Deep Reinforcement Learning (DRL) has emerged as a promising approach for designing efficient rebalancing policies because it allows for decentralised decision-making, which is particularly beneficial for AMoD services. However, evaluations have mostly focused on small-scale scenarios (Balaji et al., 2019a; Fogelson et al., 2020; Wen et al., 2017; Shi et al., 2020) and simplified networks (Mao et al., 2020), leaving a gap in scaling to larger, more complex systems. Additionally, DRL methods face challenges from non-stationarity due to concurrent learning by agents in the same environment (Laurent et al., 2011), making it difficult to ensure stable and consistent learning.

In this chapter, we explore the scalability of different DRL and MARL architectures for autonomous fleet management and present a framework based on the concept of Curriculum Learning (Bengio et al., 2009) that can efficiently expand to large networks. We propose a two-dimensional curriculum scheme that progressively augments both the number of agents and the agent's complexity, with the latter being characterised by the deep neural network architecture. Although some studies have attempted to scale MARL methods to city-sized networks, these approaches rely on contextual information like geographical data (Lin et al., 2018) and supply-demand rates (Jiao et al., 2021), which may not always be practically available.

To evaluate the proposed approach, we conduct two sets of experiments based on different adaptations of the same environment. First, we assess different DRL architectures in a small-sized case study, comparing their performance based on two fleet management metrics for a varying number of agents. Second, we compare the proposed decentralised approach with centralised baseline heuristics on a large-scale AMoD environment consisting of hundreds of autonomous vehicles (AVs) and real passenger requests sampled dynamically from the New York City (NYC) dataset (New York City Taxi and Limousine Commission, 2021).

The remainder of this chapter is organised as follows. Section 8.2 introduces the problem statement. Section 8.3 presents four different DRL learning modelling approaches, followed by the proposed efficient method for large-scale AMoD rebalancing in section 8.4. In section 8.5, a detailed description of the environment, the experimental setup, and the results are provided. Finally, in section 8.6, the main conclusions and some directions for future research are presented.

8.2 Problem Statement

We present two distinct problem formulations based on different settings, namely the single-agent and multi-agent settings. Under the single-agent formulation, all vehicle movements are governed by a centralised agent that has access to the entire global state S of the environment. In contrast, under the multi-agent formulation, each agent controls itself and has access to only partial information about the environment, which is represented by the observation O_i.

We define the individual elements as follows:

- **Agents** N: Each vehicle is considered as an individual agent, capable of being assigned to nearby orders or rebalancing to other grid locations. However, in the single-agent case, a centralised controller takes the responsibility of controlling all vehicles at any point in time, whereas, in the multi-agent formulation, each agent controls itself and has access to only partial information about the environment. Once a vehicle picks up an order, it

becomes unavailable and cannot take any further action.

- **Set of states** S **and observations** O: global state $s^t \in S$ is composed of the location of all orders and vehicles within the grid. The observation of agent i, $o_i^t \in O$, is defined by a fixed-sized region where agents have visibility of nearby orders and vehicles. This region is centred around the agent and can be of variable shape.

- **Set of actions** A: specifies the agent's movement along the grid. Individual agents take action $a_i^t \in A$. Agents can either stay in their current location or move to one of the surrounding grid locations. In the case of a ride-hailing service, agents that have already picked up a customer do not take action until they reach the customer's destination, following a previously determined shortest path. For instance, in a GridWorld scenario, each agent can move in one of the eight directions (north, south, east, west, and the four diagonals).

- **Reward function** R: We adopt a fully cooperative game setting, where all agents receive the same reward in each timestep. The reward function consists of three main components: revenue, cost, and customer satisfaction. Firstly, agents obtain a positive reward that is proportional to the trip fare earned. Secondly, agents receive a negative reward for moving to any of the neighbouring grid locations, which is proportional to the fuel cost incurred. Lastly, a penalty is assigned to missed orders that are not serviced in time, which reflects the potential revenue that could have been generated by serving the order. The reward function is presented below:

$$R = \sum_{i=1}^{N} [r_{i,fare} - r_{i,fuel} - p_{i,missed}] \tag{8.1}$$

  Where N is the number of agents, $r_{i,fare}$ is the trip fare earned by agent i, $r_{i,fuel}$ is the fuel cost incurred by the agent for moving to a neighbouring grid location, and $p_{i,missed}$ is the penalty assigned to agent i for missing orders that are not serviced in time.

- **Transition function** T: The transition function T describes how the environment changes from one state s^t to the next state s^{t+1}, given a joint action a^t taken by all

agents at time step t. In the proposed AMoD system, the transition function accounts for the availability or unavailability of vehicles at any point in time, which affects the system's state. For example, if a vehicle has just completed a trip, it becomes temporarily unavailable until the driver decides to make it available again.

8.3 DRL Architectures

In this section, we present four base DRL modelling architectures used in our experiments: Centralised Single Agent, Fully Decentralised, Parameter Sharing and Centralised Training Decentralised Execution (CTDE). Figure 8.2 presents a schematic representation of these four architectures. In the following subsections, we discuss the advantages and disadvantages of each method and its suitability for AMoD fleet control. Note that this study only considers value-based methods, and we represent the CTDE approach with the QMIX algorithm (Rashid et al., 2018a), which uses a mixing network to represent the joint action-value function.

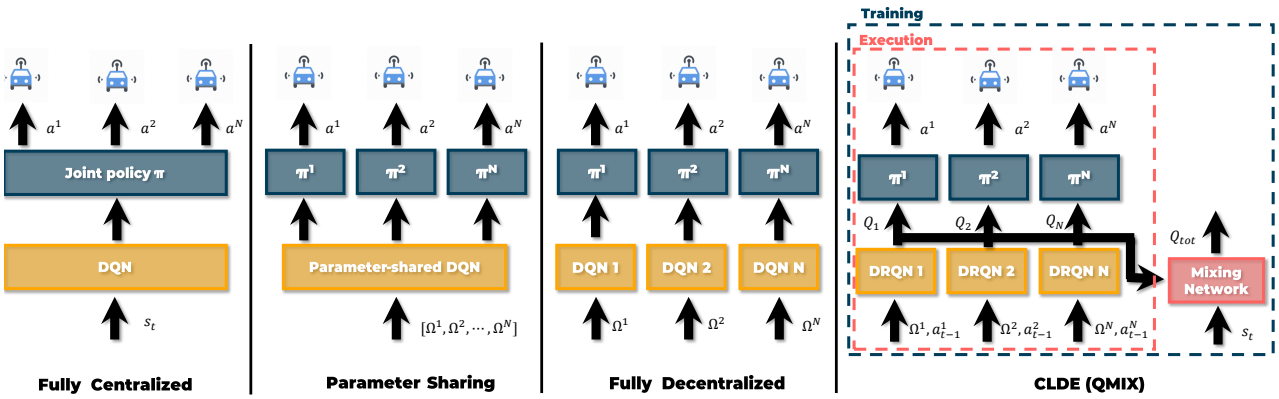


Figure 8.2: Four different DRL paradigms: Fully centralised single-agent, Parameter sharing, Fully Decentralised and Centralised Training Decentralised Execution (Represented by the QMIX algorithm).

8.3.1 Centralised Single Agent

Under this paradigm, a single controller observes a global state and takes a joint action, which triggers the environment transition. The main benefit of this approach is that the central operator has full observability and control of the fleet, overcoming potential instability issues

due to non-stationarity. However, the biggest disadvantage is that the state and action spaces grow exponentially with the number of agents. For example, a small AMoD instance with ten agents and five actions for each vehicle has a joint action space of 5^{10} combinations. Exploring an action space of this size becomes challenging, and the risk of getting trapped in local optima is higher.

8.3.2 Fully Decentralised

In the fully decentralised approach, each agent makes decisions based on its own limited view of the environment without coordinating with the others. We employ an independent Deep Q-Network (DQN) for each agent, as shown in Figure 8.2, so that every agent learns from its own observations. As a result, the total number of parameters to train increases linearly with the number of agents. While this approach allows for agents to learn completely independently, it may lead to non-stationarity in environments where coordination among agents is essential to achieving optimal performance. However, this approach does not require a central dispatcher or coordination unit, which makes it more resilient to single-point failures. Additionally, its training process is more efficient due to the reduced state space, when compared to the centralised controller. Additional details and the DQN algorithm are presented in Appendix B.

8.3.3 Parameter Sharing

In the Parameter Sharing architecture, agents still use their own observations and take their own actions; however, they share the same neural network parameters. Thus, there is a single Deep Q-network that uses the experience of all agents based on their observations, actions and rewards. In contrast to the previous approach, the number of trainable parameters is fixed and does not depend on the number of agents. Since agents under the Parameter Sharing architecture share the same neural network, there is a certain level of coordination among them. This is because all agents contribute to the updates of the shared network, allowing them to learn from each other's experiences and implicitly cooperate to achieve better performance. However, the

coordination between agents is limited to the shared parameters of the network, which might not be enough for complex tasks that require more explicit coordination mechanisms.

8.3.4 Centralised Training with Decentralised Execution

The centralised training and decentralised execution paradigm has gained popularity in the RL community, especially in MARL. This approach involves training agents offline using centralised information, such as joint observations or actions while allowing each agent to take actions based only on their own observations during execution. This promotes stability and efficiency in training as agents can learn to coordinate better with access to all relevant information while being robust to partial observability and lack of centralised coordination during execution.

Various algorithms have been proposed under this paradigm in the past few years. Examples of these are Multi-agent Deep Deterministic Policy Gradient (MADDPG) (Lowe et al., 2020), QMIX (Rashid et al., 2018a) and Counterfactual Multi-Agent (COMA) (Foerster et al., 2017). We choose the QMIX algorithm, proposed by Rashid et al. (2018a) because it allows for modelling discrete action spaces and has been shown to achieve superior performance when compared to Independent Q-learning. The details of the mathematical model and the pseudocode of QMIX are presented in Appendix B.

8.4 Curriculum-based Parameter Sharing DQN

Scaling up DRL algorithms to large environments is not trivial and requires careful tuning of the hyperparameters, architectures, and learning schedules. Learning can become unstable in such environments due to concurrent learning by agents. Achieving effective learning and agent coordination is still an active field of research. To address this, different methods have been proposed, such as importance sampling (Foerster et al., 2018), fingerprints (Foerster et al., 2018), and Curriculum Learning (Wang et al., 2019). Among these, Curriculum Learning is particularly promising, as it allows for faster learning by introducing a curriculum of increasingly

complex tasks (Bengio et al., 2009).

We propose a method based on Curriculum Learning (CL) to help stabilise the learning procedure for large-scale autonomous fleet rebalancing. The core idea behind CL is to generate a training strategy that begins with simple tasks and gradually increases the difficulty until the full task is finally reached. CL in RL has been mostly used for single-agent tasks, although a few studies have proposed CL techniques for MARL (Agarwal et al., 2019; Wang et al., 2019). Particularly, population curriculum (Wang et al., 2019) has shown to improve performance and algorithm stability in large-scale multi-agent problems. Population curriculum begins the training phase with a low number of agents and progressively adds new agents until the desired amount is reached. Different knowledge transfer mechanisms can be applied during the training phase to guarantee stability and speed up learning.

The proposed method, PSDQN-CL, features a two-dimensional curriculum scheme to facilitate stable and efficient learning for large-scale autonomous fleet rebalancing. The first dimension is a multi-agent curriculum scheme $\mathcal{C}$, which varies the number of agents in consecutive episodes, starting with a small number of agents and gradually increasing it until the desired amount is reached. This population curriculum approach, inspired by previous studies (Agarwal et al., 2019; Wang et al., 2019), has shown promising results in improving algorithm stability and performance in large-scale multi-agent problems.

The second dimension of the curriculum is achieved by gradually increasing the complexity of the agents. In other words, simpler agents are used at the beginning of the curriculum, and more complex agents are introduced as the training progresses. This complexity is represented by the deep neural network architecture, which evolves from shallower and simpler networks to deeper and more sophisticated ones. This allows agents to gradually learn and master skills necessary for successful autonomous fleet rebalancing without being overwhelmed by the complexity of the full task from the beginning. The task gets increasingly more complex as the training progresses by progressively introducing more customer requests, and the agents need to be able to handle more complex scenarios and make more difficult decisions.

To facilitate knowledge transfer and increase learning efficiency, the proposed method includes

three knowledge transfer mechanisms within the curriculum scheme. These mechanisms are buffer reuse, model reload, and knowledge distillation. By sharing the replay buffer and reusing the same network parameters when training a new task, the agents can learn from previous experiences more efficiently, leading to faster and more stable learning. Additionally, the knowledge distillation method encourages knowledge transfer between agents of different complexity, ultimately leading to the final policy that can efficiently perform the full task with a large number of agents.

We adopt policy distillation over Q-values (Rusu et al., 2016), or Q-distillation, to transfer knowledge from simpler, trained Q-functions to more complex Q-functions as the curriculum progresses. This process allows the more complex agents to benefit from the knowledge acquired by simpler agents, thereby improving learning efficiency and stability.

Let $Q_{\theta_k}(s, a)$ represent the Q-function of the agent at the k-th stage of the curriculum, where θ_k are the parameters of the Q-network at that stage. The Q-distillation process minimises the difference between the Q-values of the teacher network (simpler, trained Q-function) and the student network (more complex Q-function being trained). The distillation loss $L_{distill}$ is formulated as follows:

$$L_{distill} = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\left(Q_{\theta_k}(s,a) - Q_{\theta_{k-1}}(s,a)\right)^2\right],$$

where $Q_{\theta_k}(s, a)$ is the Q-value predicted by the student network, $Q_{\theta_{k-1}}(s, a)$ is the Q-value predicted by the teacher network, and $\mathcal{D}$ is the distribution of state-action pairs sampled from the replay buffer.

The total loss function for training the student network combines the standard DQN loss L_{DQN} with the distillation loss $L_{distill}$:

$$L_{total} = L_{DQN} + \lambda L_{distill},$$

where λ is a hyperparameter that controls the trade-off between the standard DQN loss and

the distillation loss.

The simpler agents are represented by a regular DQN architecture with one hidden layer with 128 hidden units. As the curriculum progresses, the complexity of the agents rises by increasing the number of hidden layers and units. The highest complexity is represented by a network with three hidden layers.

The proposed Curriculum-based Parameter-sharing DQN algorithm is presented in Algorithm 3. The multi-agent population is updated according to the curriculum scheme during every global step until the desired amount of agents N_d is reached. New agents that are created share the same network weights and the experience replay buffer with agents that were created in previous steps. The demand pattern is also updated according to a demand scheme so that when more agents are introduced to the environment, there are progressively more customer requests. Additional implementation details and the main hyperparameters of PSDQN-CL are provided in Appendix A.

8.5 Experiments

This section describes the two sets of experiments we conducted. The first set investigates the scalability of the various DRL paradigms presented in section 8.3 within a small GridWorld. The second set assesses the effectiveness of the proposed method compared to a series of baselines on a real-world instance of an AMoD system that uses the Manhattan network and the NYC taxi dataset (New York City Taxi and Limousine Commission, 2021). The latter experiment is particularly challenging due to the large number of agents learning simultaneously in the same environment.

The two sets of experiments share the same core logic for the environment. In this environment, there are multiple vehicles and dynamically generated order requests. Vehicles have the option to either match orders or move to surrounding grid locations. When a vehicle matches an order, it must take the passenger to their destination and remains inactive during this time. After dropping off the passenger, the vehicle becomes available once again and can continue taking

Algorithm 3 Curriculum-based Parameter Sharing DQN (PSDQN-CL)

Initialise replay buffer $\mathcal{D}$
Initialise Q-networks $Q_{\theta_0}, Q_{\theta_1}, \ldots, Q_{\theta_N}$ for N stages of curriculum
Initialise number of agents $n = 1$
Initialise demand pattern for customer requests
for each curriculum stage $k = 1, 2, \ldots, N$ **do**
    Initialise Q-network Q_{θ_k} for the current stage
    for each episode e **do**
        for each step t **do**
            Select actions $a_1^t, a_2^t ... a_N^t$ according to an ϵ-greedy policy based on Q_{θ_k}
            Execute joint action a^t and observe reward r^t and $o^{t+1} = (o_1^t, o_2^t, ..., o_N^t)$
            Store transition (o^t, a^t, r^t, o^{t+1}) in replay buffer $\mathcal{D}$
            Sample a random minibatch of transitions (o, a, r, o') from $\mathcal{D}$
            Compute target Q-value: $y = r + \gamma \max_{a'} Q_{\theta_k}(o', a')$
            Compute standard DQN loss: $L_{DQN} = \mathbb{E}\left[(y - Q_{\theta_k}(o, a))^2\right]$
            if $k > 0$ **then**
                Compute distillation loss: $L_{distill} = \mathbb{E}\left[\left(Q_{\theta_k}(o, a) - Q_{\theta_{k-1}}(o, a)\right)^2\right]$
                Compute total loss: $L_{total} = L_{DQN} + \lambda L_{distill}$
            else
                $L_{total} = L_{DQN}$
            end if
            Perform a gradient descent step to minimise L_{total} and update parameters θ_k
        end for
        if population curriculum condition is met **then**
            Increment number of agents n
            Initialise new agents with shared network weights and replay buffer
        end if
        Update demand pattern according to demand scheme
    end for
end for
Evaluate the performance of the final Q-network Q_{θ_N} on the full task with the desired number of agents N_d

action. The paths from the customer's origin to their destination, as well as their total length, have been predetermined using a shortest path algorithm. The implementation details can be found in Appendix A.

In these experiments, we simplify the AMoD environment using a grid-based setup for dispatching and rebalancing. This abstraction allows us to focus on the scalability challenges in DRL for fleet management without the added complexity of modelling a full road network. The controlled GridWorld setting supports clear benchmarking of DRL paradigms, enabling a focused examination of stability and learning performance within a foundational structure.

This approach, in line with prior AMoD studies Gueriau and Dusparic (2018); Lin et al. (2018), prioritises computational efficiency, which is especially important when scaling to large agent populations.

Both the initial GridWorld and the Manhattan experiment—based on NYC taxi data—use a grid abstraction for vehicle movement and service areas, simplifying the spatial representation while retaining core aspects of the dispatching and rebalancing tasks. This setup allows us to establish baseline insights into DRL scalability and coordination under structured but simplified spatial constraints, providing a foundation for further studies that may integrate more intricate network representations.

To compare the different methods, we focus on two fleet management metrics:

- Empty vehicle mileage (EVM): this accounts for the total distance travelled by the vehicles when they are not carrying a customer. Ideally, the fleet operator should minimise empty vehicle mileage by continuously rebalancing the vehicles to zones with higher demand.
- Average customer waiting time (AWT): the fleet operator should reduce overall customer waiting times, as this is directly related to customer satisfaction. Waiting time is measured from the moment the customer requests a vehicle until it is picked up.

In addition to the fleet management metrics, we compare the performance of the different algorithms based on their learning curves, i.e., the total reward over a fixed number of episodes. For each experiment, we average the performance over ten independent runs with different seeds.

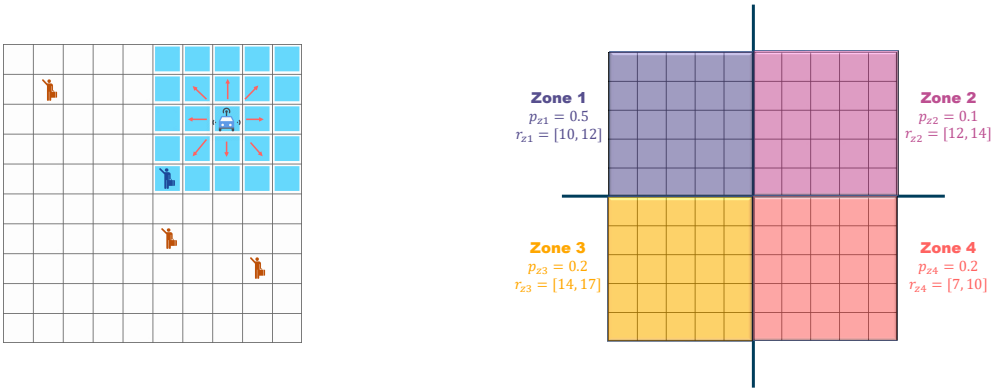
8.5.1 GridWorld environment

The first experiment focuses on exploring the different DRL modelling architectures in a controlled environment with low non-stationarity. The rebalancing environment is modelled as a GridWorld, where agents can take actions to move directly to their neighbouring grids and can match customers in real-time. The GridWorld is 10x10 in size and contains up to five agents

and five customers at any given time. The partial observability of the agents is represented as a 5×5 view of other orders and agents, as shown in figure 8.3 (a). In case an agent reaches the edge of the environment and chooses an action that takes it off the grid, it will stay in the same position with no penalty. To encourage agents to match customers quickly, a penalty reward is established for orders that are not matched within a certain period of time.

To simulate the heterogeneous nature of customer demand patterns in real cities, the GridWorld environment is divided into four distinct zones, each with different order generation and fare reward probabilities. Agents are expected to learn how to rebalance to zones with more orders and/or higher reward. Orders are generated based on a constant probability and assigned to one of the four zones with the following discrete probability distribution: $p_{z1} = 0.5, p_{z2} = 0.1, p_{z3} = 0.2, p_{z4} = 0.2$. To account for the different reward probabilities of each zone, a truncated normal distribution is used with minimum and maximum values for each zone: $r_{z1} = [10, 12], r_{z2} = [12, 14], r_{z3} = [14, 17], r_{z4} = [7, 10]$.

Each zone's parameters are also summarised in figure 8.3 (b).



(a) Agent and order representations (b) Zone-specific parameters

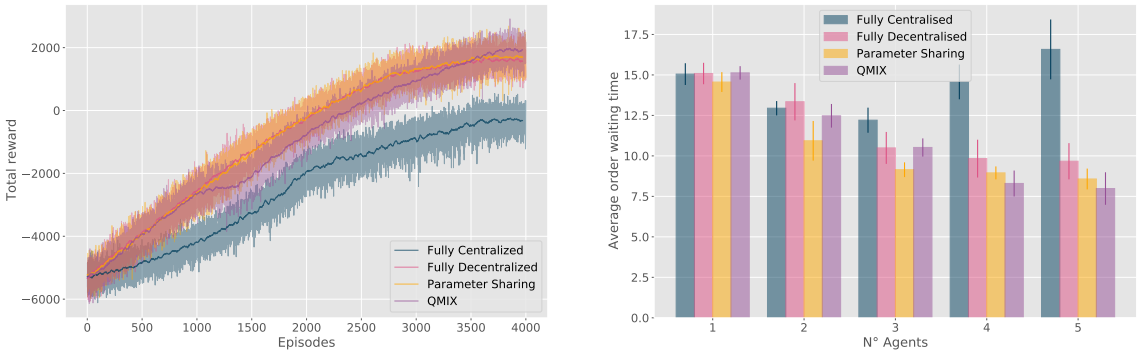
Figure 8.3: GridWorld environment. (a) Agent and order representations. (b) Zone-specific parameters.

Comparison between centralised and decentralised learning.

We performed a comparison of four different DRL modelling architectures: Fully Centralised, Fully Decentralised, Parameter-Sharing and CTDE. First, we compared the total reward of

each method over 4000 episodes. We considered different numbers of agents for each run in order to assess the scalability of each method's performance. Second, we evaluated the final control policies based on the total EMV and the average CWT.

Table 8.1 summarises the results for the GridWorld case study. It contains the final reward, empty mileage ratio and average waiting time for all the experiments. For 1 and 2 agents, the final total reward is similar for all four architectures. However, the performance of the Fully Centralised architecture degraded substantially when increasing to 3, 4 and 5 agents. This is also reflected in the empty mileage ratio and the average waiting time of the final policies. A comparison of the learning curves (total reward per episode) for the case with five agents is shown in figure 8.4 (a). As shown, the fully centralised approach systematically converges to a lower total reward than that of the other three methods. This can be explained by the increased complexity of the joint action and state spaces.



(a) Learning curves for each method (5 agents) (b) Passenger average waiting time for each method and different number of agents

Figure 8.4: Results for the GridWorld case study. (a) Learning curves for each method (5 agents). (b) Passenger average waiting time for each method and different number of agents.

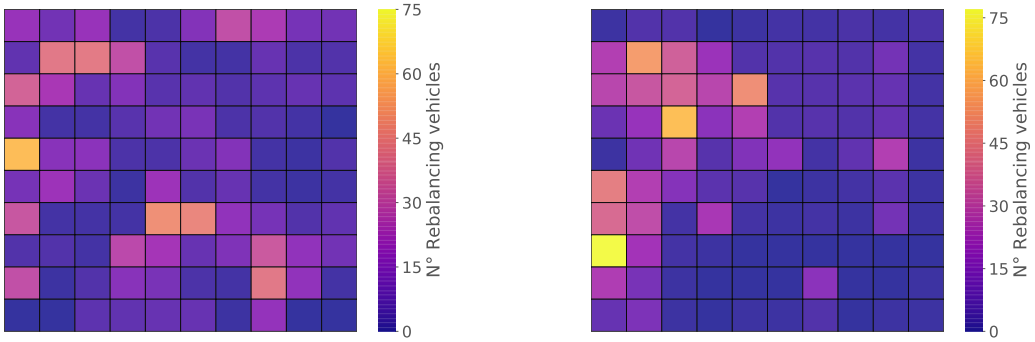
The results of the comparison between the four different DRL architectures show that QMIX outperformed all other methods in terms of total reward, except for the scenario with only one agent. The difference between QMIX and the other methods was more evident when increasing the number of agents. These findings suggest that centralised information (joint value function) has a positive effect on algorithm performance in small-scale multi-agent scenarios. However, it is important to consider the trade-off between performance and computational complexity of obtaining and using the full state.

Table 8.1: Performance comparison of the different methods for the GridWorld case study.

N° agents	Method	Total Reward		EVM ratio		AWT	
		Avg.	Std.	Avg.	Std.	Avg.	Std.
1	Fully Centralised	161.43	11.91	0.45	0.02	15.05	0.67
	Fully Decentralised	157.48	19.71	0.49	0.04	15.09	0.66
	Parameter Sharing	**170.21**	**10.91**	**0.46**	**0.01**	**14.56**	**0.61**
	QMIX	140.42	15.72	0.43	0.03	15.13	0.42
2	Fully Centralised	391.93	66.19	0.46	0.03	12.94	0.44
	Fully Decentralised	505.96	41.91	0.48	0.04	13.34	1.15
	Parameter Sharing	445.69	76.28	0.48	0.04	10.93	1.22
	QMIX	**533.62**	**86.72**	**0.44**	**0.04**	**12.48**	**0.73**
3	Fully Centralised	381.31	283.94	0.52	0.02	12.20	0.78
	Fully Decentralised	917.03	219.71	0.46	0.03	10.49	0.99
	Parameter Sharing	929.87	180.59	0.50	0.02	9.15	0.46
	QMIX	**1143.01**	**287.32**	**0.45**	**0.03**	**10.52**	**0.56**
4	Fully Centralised	5.11	195.47	0.63	0.03	14.57	1.07
	Fully Decentralised	1144.09	226.37	0.46	0.02	9.83	1.17
	Parameter Sharing	1290.60	192.86	0.47	0.03	8.95	0.40
	QMIX	**1417.88**	**261.40**	**0.46**	**0.01**	**8.30**	**0.80**
5	Fully Centralised	−95.93	412.55	0.77	0.10	16.58	1.85
	Fully Decentralised	1316.17	357.81	0.50	0.04	9.67	1.12
	Parameter Sharing	1548.59	503.03	0.46	0.01	8.57	0.64
	QMIX	**1992.11**	**190.79**	**0.43**	**0.01**	**7.98**	**1.00**

Figure 8.4 (b) shows the average passenger waiting time obtained after executing the final control policies for each method. When increasing the number of agents, all methods except the fully centralised have a downward trend regarding the average waiting time. For the cases with 4 and 5 agents, the centralised approach is not able to learn good performing policies. Vehicles tend to move sub-optimaly and, thus, end up arriving late or simply losing customers. The other three methods achieve similar average waiting times, although the QMIX algorithm slightly outperforms the other two in the cases with 4 and 5 agents.

Figure 8.5 provides further insight into the behaviour of the fully decentralised and QMIX methods. The heatmaps illustrate the rebalancing actions taken by the final control policies obtained by each method. As shown, the QMIX algorithm's policy continuously rebalances vehicles towards zones 1 and 3, which have the highest order generation probability and the highest reward per order, respectively. In contrast, agents trained with the fully decentralised approaches tend to remain in the same zone and rebalance to high reward zones less frequently. These results demonstrate the effectiveness of the QMIX algorithm in learning efficient control policies in small-scale environments.



(a) Fully decentralised method (b) QMIX algorithm

Figure 8.5: Heatmaps of rebalancing actions obtained with the (a) Fully decentralised method (b) QMIX algorithm.

8.5.2 Manhattan network

One of the biggest challenges in fleet management problems is to design methods that scale to city-sized instances (Schwarting et al., 2018). In such environments, MARL methods can be largely affected by non-stationarity. Each agent's perspective on the environment constantly changes as other agents adapt their policies. This causes instability issues in MARL methods, leading to poorly performing multi-agent policies. Algorithms that use centralised information during training, such as the QMIX algorithm, should perform better in this type of environment. This is because the agents have access to other agents' value functions during training; thus, from their perspective, there are no unexpected changes in the environment (Papoudakis et al., 2019).

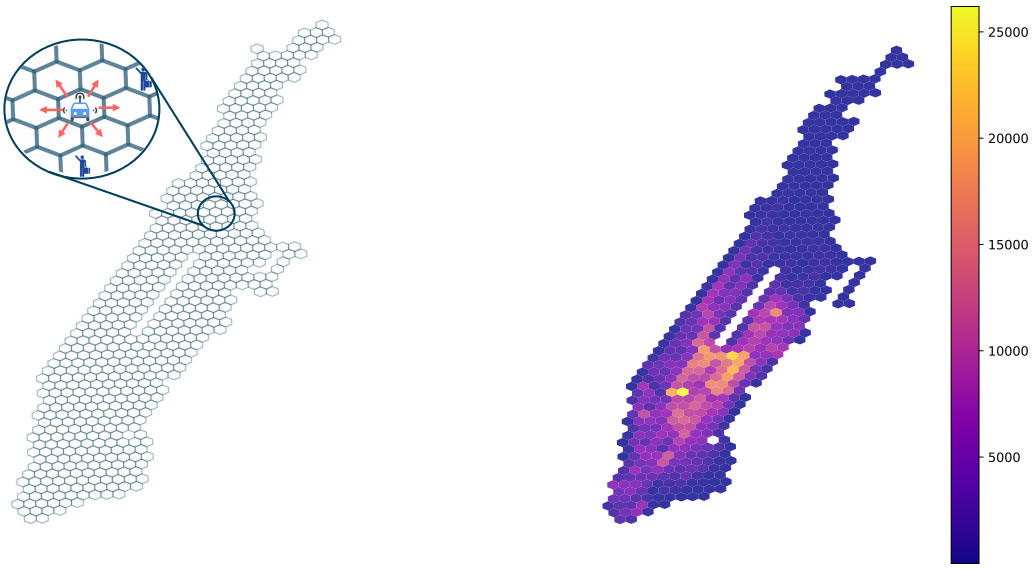
Our approach to addressing non-stationarity in large-scale environments adopts a two-dimensional Curriculum Learning scheme that considers the number of agents and the complexity of their policies. We also incorporate three knowledge transfer mechanisms to expedite and stabilise the learning process. In the following subsections, we benchmarked our method against centralised control baselines and the QMIX algorithm but excluded fully centralised and fully decentralised architectures due to scalability concerns in large-scale settings.

The evaluation of the different methods involved an environment based on the Manhattan road network, which accounts for real-time interactions between thousands of customer requests and

vehicles. The network comprises 4480 nodes and 9750 edges and is depicted in Figure 8.6 (a).

To formulate the problem as a Dec-POMDP, we used the H3 geospatial indexing system developed by Uber (Brodsky, 2018) and partitioned the network into hexagonal grids. To determine the grid size, we conducted a comprehensive preliminary experiment comparing a random rebalancing heuristic and PSDQN for different grid sizes. We selected the grid size based on the biggest performance gap observed between the two methods. This optimised the simulation parameters for effective rebalancing without unnecessary vehicle movement in the compact Manhattan network..

We implement a two-step matching heuristic, similar to the approach utilised in (Lin et al., 2018). Initially, orders are assigned to available agents within the same grid, considering order time and price. Subsequently, any unassigned orders are then allocated to available agents in adjacent grids. This process guarantees sensible matches between customers and AVs in relation to proximity and priority. Should an order still remain unassigned following the second step, it will wait for subsequent matching rounds.



(a) Agent and order representations (b) Passenger pick-up data (one month)

Figure 8.6: Manhattan environment. (a) Agent and order representations. (b) Passenger pick-up data (1 month).

To simulate a realistic scenario, we used the NYC Taxi Dataset to generate customer demands for the Manhattan area during weekday evening peak hours (4 P.M. to 8 P.M.). The dataset

provides pick-up and drop-off information for yellow taxi trips in New York City (New York City Taxi and Limousine Commission, 2021). The spatial distribution of customer requests within the network is shown in the order pick-up heatmap in Figure 8.6 (b), and Figure 8.7 presents the number of orders picked up during the four-hour period on each of the 40 different weekdays used in our experiments. Our dataset consisted of over 2 million pick-up data points, providing a comprehensive and realistic basis for our simulations. Each simulation episode lasts 400 timesteps, representing the full four-hour period, and each timestep corresponds to 36 seconds in the real world.

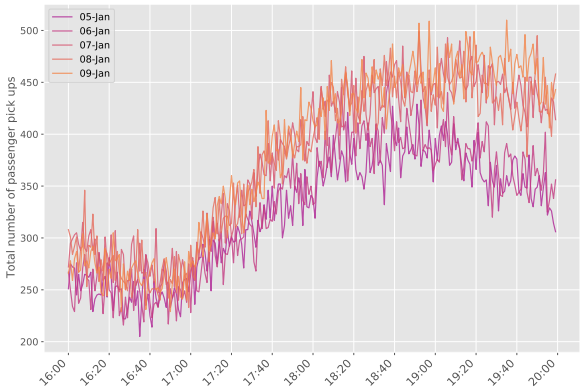


Figure 8.7: Total number of passenger pick-ups between 4 and 8 p.m. on five different weekdays.

Baselines

The performance of the proposed model is compared against baseline DRL models and four traditional centralised baselines:

- Diffusion: randomly rebalances agents to nearby zones and serves as the most basic baseline that learning algorithms should consistently outperform.
- Greedy Heuristic: this method makes local optimal decisions at each step by moving agents towards the neighbouring region with the highest number of orders. This method is considered naive due to its lack of memory.
- Tabular Q-learning: the standard form of the Q-learning algorithm, where Q-values are stored in a table and updated according to the Bellman equation. An $\epsilon - greedy$ policy is used. This method is often used as a fleet rebalancing baseline.

- Fluidic optimal policy: this method minimises the average number of rebalancing vehicles in a network while maintaining stability. It was introduced by (Pavone et al., 2012) and has shown superiority over common rebalancing heuristics (Hörl et al., 2019). The approach involves solving a linear program that minimises the sum of travel times weighted by rebalancing rates (α_{ij}). The linear program formulation corresponds to a minimum cost flow problem:

$$\begin{aligned} \min \quad & \sum_{i,j} T_{ij}\alpha_{ij} \\ \text{s.t.} \quad & \sum_{i \neq j}(\alpha_{ij} - \alpha_{ji}) = -\lambda_i + \sum_{j \neq i} \lambda_j p_{ij} \qquad \forall i \in V \\ & \alpha_{ij} \geq 0 \qquad\qquad\qquad\qquad\qquad\quad \forall i, j \in V \end{aligned} \tag{8.2}$$

  where T_{ij} corresponds to the travel time between node i and node j, α_{ij} is the rate at which node i sends empty vehicles to node j, λ_i is the customer arrival rate at node i and p_{ij} is the fraction of customers at node i whose destination is node j. To implement this model in the Manhattan environment, we solve the LP problem at the beginning of each episode and update the rebalancing rates α_{ij} on each timestep. For this, we assume that the passenger arrival rates λ_i and demand pattern rates p_{ij} are completely known beforehand (model-based). Note that this is not the case for the DRL methods, as they learn in real-time, with no previous knowledge of the demand (model-free).

Performance comparison.

Figure 8.8 shows the impact of multi-agent Curriculum Learning (CL) on the learning process as a whole. This Figure demonstrates the benefits of using curricula in large-scale multi-agent settings. While the no-curriculum method can achieve high-performing policies during training, it consistently converges to low-valued policies. The QMIX algorithm, which uses the centralised value function during learning, faces similar difficulties, with agents unable to learn consistently. In some cases, overall performance deteriorates over time, causing agents to forget what they learned in previous episodes. Conversely, when employing the curriculum,

the average total reward steadily increases with the number of episodes, although some poorly performing episodes may still occur during training.

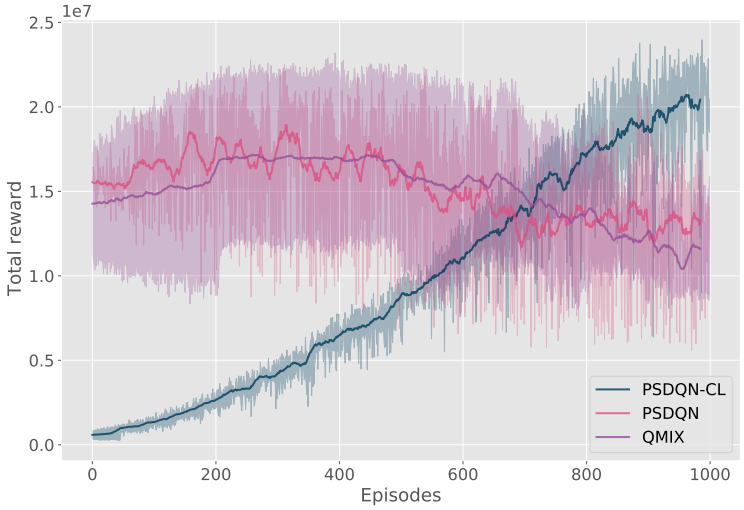


Figure 8.8: Comparison of the learning curves for the Manhattan environment (250 agents).

Table 8.2 summarises the results for the Manhattan case study. The performance of all learning methods, as well as all the baselines, are presented. The evaluation encompasses four distinct problem scales, ranging from scenarios involving 250 to 1000 agents. The proposed method, PSDQN-CL, outperformed all heuristics and state-of-the-art RL methods after 1000 training episodes. This superiority extends to the fleet management metrics, where the PSDQN-CL method consistently outperforms across all agent counts.

As the number of agents increases, traditional heuristics like Diffusion and Greedy exhibit diminishing effectiveness. This is consistent with expectations, as these methods struggle to adapt to the escalating complexity and interdependencies inherent in larger agent populations. Conversely, PSDQN and QMIX show more robust performance, which can be attributed to their capacity for learning intricate coordination strategies. However, even these methods suffer from stability challenges and, consequently, fall short of surpassing the performance of the fluidic optimal policy.

Figure 8.9 provides a visual representation of the heatmap of rebalancing actions for each trained policy, including the proposed method, the Fluidic Optimal Policy, and the QMIX algorithm. By showing the distribution of rebalancing actions in different zones of the AMoD system, the

Table 8.2: Performance comparison of the different methods for the Manhattan case study.

N° agents	Method	Total Reward ($\times 10^7$)		EVM ratio		AWT	
		Avg.	Std.	Avg.	Std.	Avg.	Std.
250	Diffusion	1.20	0.12	0.29	0.02	10.11	0.35
	Greedy	1.71	0.10	0.21	0.04	8.32	0.41
	Fluidic policy	1.80	0.25	0.16	0.03	8.21	0.40
	Tabular Q-learning	1.55	0.15	0.24	0.02	8.43	0.33
	QMIX	1.37	0.28	0.21	0.03	8.45	0.30
	PSDQN	1.42	0.35	0.24	0.04	8.58	0.26
	PSDQN-CL	**2.10**	**0.31**	**0.13**	**0.04**	**7.40**	**0.31**
400	Diffusion	3.15	0.15	0.42	0.03	9.75	0.28
	Greedy	3.51	0.20	0.40	0.03	8.40	0.15
	Fluidic policy	4.03	0.22	0.37	0.04	6.66	0.28
	Tabular Q-learning	3.30	0.46	0.42	0.05	9.29	0.29
	QMIX	3.55	0.29	0.38	0.03	8.44	0.23
	PSDQN	3.43	0.32	0.38	0.04	8.96	0.33
	PSDQN-CL	**4.37**	**0.37**	**0.34**	**0.03**	**5.75**	**0.34**
600	Diffusion	4.44	0.08	0.49	0.10	9.99	0.45
	Greedy	4.72	0.25	0.48	0.04	8.01	0.23
	Fluidic policy	5.23	0.18	0.37	0.06	5.71	0.12
	Tabular Q-learning	4.41	0.67	0.44	0.06	9.13	0.44
	QMIX	4.86	0.32	0.44	0.10	7.26	0.46
	PSDQN	4.82	0.15	0.46	0.07	8.00	0.24
	PSDQN-CL	**5.96**	**0.48**	**0.34**	**0.08**	**4.65**	**0.27**
1000	Diffusion	5.54	1.23	0.51	0.67	9.56	0.97
	Greedy	5.45	0.44	0.50	0.13	8.25	0.57
	Fluidic policy	7.27	0.87	0.34	0.09	5.12	0.54
	Tabular Q-learning	5.89	0.88	0.43	0.09	6.17	0.69
	QMIX	6.27	0.86	0.34	0.08	7.21	0.61
	PSDQN	5.92	0.56	0.37	0.12	6.64	0.55
	PSDQN-CL	**7.97**	**0.62**	**0.29**	**0.12**	**4.34**	**0.42**

heatmaps offer insights into the effectiveness of the proposed method compared to state-of-the-art algorithms. Specifically, the results highlight the percentage of total rebalancing actions allocated to each grid.

The examination of the rebalancing heatmaps for a scenario with 400 agents reveals that the QMIX algorithm encounters difficulties in effectively redistributing vehicles towards regions with higher numbers of pickups. The pickup distribution across space is presented in Figure 8.6 for reference. Conversely, the proposed PSDQN-CL method captures the demand pattern effectively and rebalances vehicles to the zones with higher pickups. The Fluidic Optimal Policy performs better than the QMIX algorithm but still rebalances a significant number of vehicles to the northern zone, where pickups are low.

The rebalancing policies of the QMIX algorithm for 600 and 1000 agents show considerably worse performance, with a high number of vehicles being rebalanced to the northern region, which can be in part attributed to the increased non-stationarity of the environment. In contrast, the PSDQN-CL method continues to effectively capture the demand pattern and

provide a high-performing rebalancing policy. These results demonstrate the superiority of the proposed method over state-of-the-art algorithms in allocating rebalancing actions in different zones of the AMoD system, especially in environments with many agents.

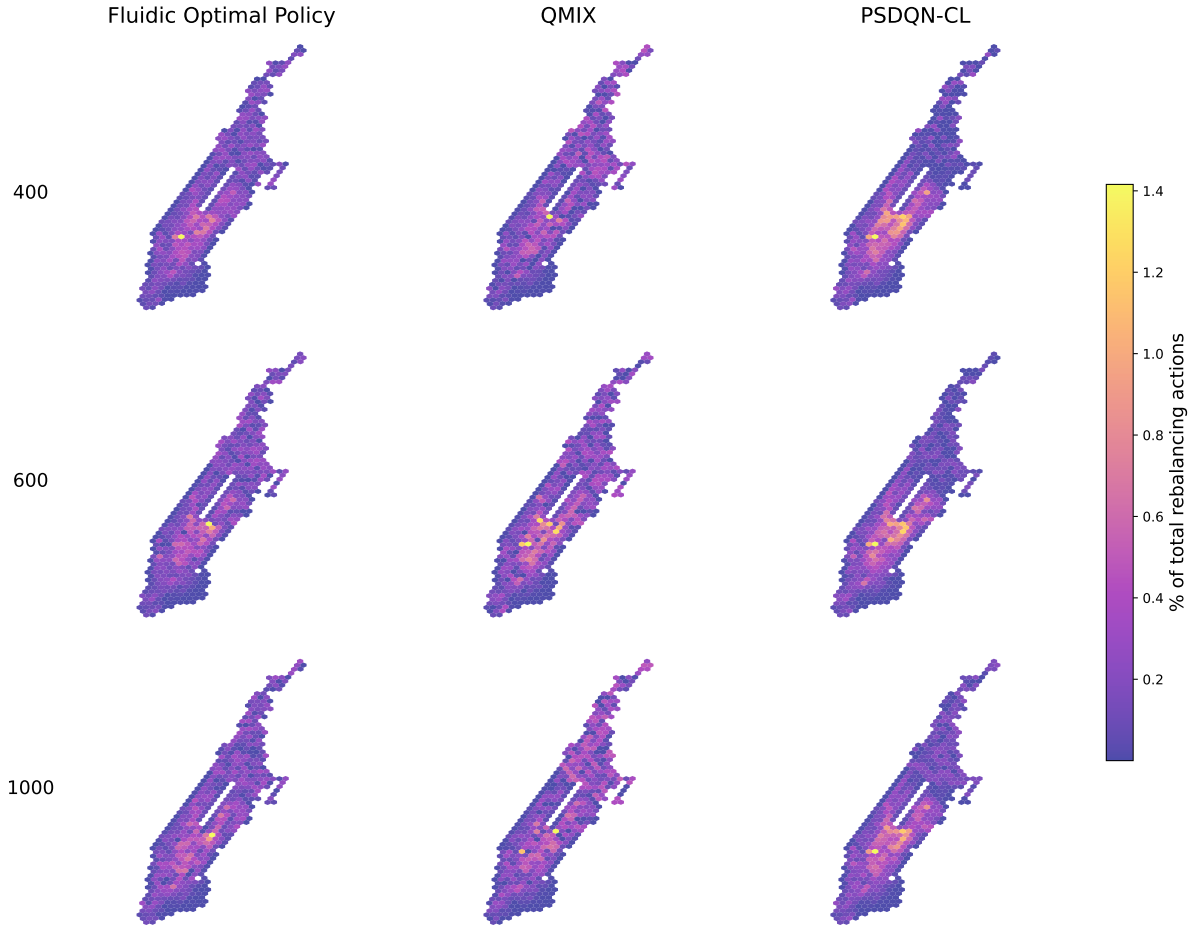


Figure 8.9: Heatmaps of rebalancing actions for the PSDQN-CL method as well as the Fluidic Optimal Policy and the QMIX algorithm. The results shown correspond to percentages of total rebalancing actions.

Figure 8.10 shows the average training time for each of the deep learning methods. As shown, PSDQN-CL is substantially faster than the other two learning methods. The average training time per episode is 73% and 47% less when compared with the QMIX and the base PSDQN algorithm (Avg. over the four scenarios), respectively. This is due to the progressive training of agents given by the multi-agent curriculum scheme. The QMIX method becomes computationally intensive for these large scenarios, mainly because of the extra resources needed for the mixing network during training.

The PSDQN-CL method demonstrates fast execution time, taking less than 0.01 seconds to

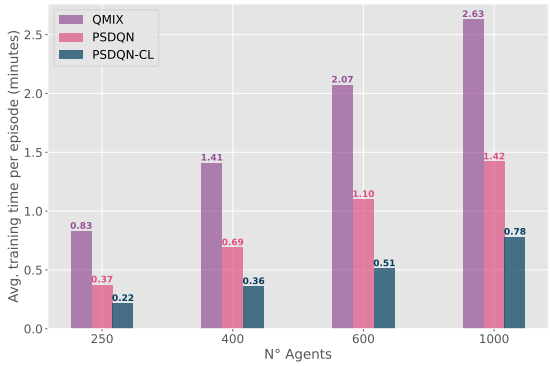


Figure 8.10: Average training time per episode for the three learning methods.

evaluate actions for all 1000 vehicles, making it a viable option for offline training and real-time implementation. Moreover, the algorithm's execution time is independent of the number of agents, making it practical to implement for very large fleet sizes, such as 10000 vehicles, although the training time may increase due to higher simulator complexity.

8.6 Concluding Remarks

This chapter addresses a problem at the strategic decision-making level, focusing on optimising large-scale autonomous fleet management. Deep Reinforcement Learning has emerged as an effective approach for managing the complexities of such systems. However, such large environments suffer from non-stationarity, which can heavily undermine the training procedure, leading to poorly performing policies. In this chapter, we proposed an efficient MARL method that can guarantee learning stability for large-scale AMoD systems. This method is based on multi-agent curricula and knowledge transfer mechanisms to help stabilise the training phase. We developed two case studies. First, we systematically compared centralised and decentralised learning methods in a controlled small-scale environment. Second, we assessed the benefit of using the proposed curriculum-based method, comparing it against state-of-the-art centralised and DRL methods in a city-sized AMoD environment.

The results demonstrated that the fully centralised approach consistently underperforms decentralised methods, even in small environments with as few as five agents. The performance gap

becomes more significant when increasing the number of agents due to the exponential increase in state and action spaces. Fully decentralised methods perform well; however, it is important to consider the potential for prohibitively large network parameters. Additionally, while centralised information during training may provide some benefit in small-scale environments, this must be weighed against the computational burden of retrieving and using the centralised value function.

The superiority of the proposed curriculum-based scheme over all baselines has been demonstrated, outperforming QMIX in environments with a large number of agents. The proposed method showed an average improvement in total reward of 31% and 12% when compared against the QMIX algorithm and the Fluidic Optimal Policy, respectively. It is worth noting that environments with over 100 agents become highly non-stationary, and as a consequence, decentralised learning algorithms struggle to learn consistently, even when agents have access to centralised information (such as the QMIX algorithm). The heatmaps of rebalancing actions reveal that the PSDQN-CL method effectively captures the demand pattern and rebalances vehicles to the zones with higher pickups, while the QMIX algorithm struggles to learn consistently. The results also demonstrate the effectiveness of using curricula in large-scale multi-agent settings, as the average total reward steadily increases with the number of episodes when employing the curriculum.

The results of our study indicate that curriculum learning can provide a significant boost in learning speed, with an average speed-up of 47% compared to the baseline parameter-sharing DQN. However, we believe that further research should explore other methods of stabilising learning in large-scale environments and compare their speed and effectiveness. Recent efforts in this direction include the use of experience replay trajectories (Omidshafiei et al., 2017), temporal fingerprints (Foerster et al., 2018), and training agents with policy ensembles (Lowe et al., 2020). These approaches have shown promise in improving the stability and effectiveness of multi-agent learning, and they should be considered as potential alternatives or complements to curriculum learning in future research.

Chapter 9

Conclusions and Future Work

This thesis has explored the multifaceted challenges and opportunities presented by Multi-Agent Reinforcement Learning in the context of Autonomous Vehicles. Throughout the research, we have addressed key challenges at each of the three decision-making levels—strategic, tactical, and operational. By utilising MARL, this work has provided solutions that enhance coordination, communication, scalability, safety, and the effective transfer of policies from simulation to real-world environments.

At the operational level, as covered in Chapter 4, the focus was on sim-to-real transfer to ensure that policies trained in simulation could be deployed in dynamic real-world environments. Techniques introduced here addressed the challenges of robustness and uncertainty in low-level control tasks like trajectory planning and obstacle avoidance.

At the tactical level, three methodologies were developed to address medium-term decision-making. Chapter 5 introduced a method for safe manoeuvring in the presence of emergency vehicles, enhancing coordination in critical traffic scenarios. Chapter 6 focused on communication strategies, enabling AVs to share perception features and convey vehicle intent for effective navigation in complex urban environments. Chapter 7 examined adversarial threats to V2X communication, presenting a framework for cooperative defence to safeguard against attacks and ensure communication integrity in tactical decision-making.

At the strategic level, this research focused on managing autonomous fleets in the context of large-scale mobility services, addressing the key challenges of non-stationarity, fleet-wide coordination, and scalability. Chapter 8 examined the scalability of MARL to manage complex fleets of autonomous vehicles, optimising vehicle relabalancing and reducing passenger wait times.

In this final chapter, we first summarise the key findings and explain how the three objectives were addressed (section 9.1). We then discuss the broader impact and practical applications of the methods and solutions presented in this thesis for the AV industry (section 9.2). Finally, we identify the challenges that lie ahead and propose strategies to tackle them (section 9.3).

9.1 Summary of Key Findings and Contributions

This thesis systematically addressed the research objectives outlined in Chapter 1, demonstrating significant advancements in the field of autonomous driving through Multi-Agent Reinforcement Learning. The key findings and how each research objective was met are summarised below:

1. **Coordination and Safety:** Coordination and safety across decision-making levels were addressed, from low-level control to higher-level manoeuvres. At the low level, a domain randomisation approach was applied to enhance the transferability of simulation-trained policies to real-world environments, improving robustness in dynamic, unpredictable settings. At the higher level, a MARL-based coordination method was developed, combined with distance-based risk modelling, to facilitate safe manoeuvring in complex emergency vehicle scenarios.

2. **Secure and Efficient Communication:** Robust communication frameworks were developed to enable secure and efficient collaboration among AVs. First, two communication strategies were introduced: one for sharing vehicle intent through an observation prediction module and another for using compressed LiDAR features to facilitate real-time data

sharing. These strategies enhanced coordination and ensured safe navigation in challenging urban scenarios. Second, a cooperative defence framework was developed to protect against adversarial communication attacks. This framework demonstrated near-perfect zero-shot performance in maintaining network cooperation and robustness against covert attacks, where adversarial training of MARL algorithms failed.

3. **Scalability:** Scalable MARL control mechanisms were developed for large-scale autonomous vehicle systems. A curriculum-based MARL method was introduced, specifically tailored for Autonomous Mobility-on-Demand (AMoD) systems. This approach demonstrated superior stability and efficiency in city-scale simulations, effectively addressing the challenges of non-stationarity and coordination among hundreds of agents. We showed that non-stationarity can significantly impair the performance of traditional methods in environments where hundreds of agents learn simultaneously.

In the next subsection, we discuss the significance of these findings and their impact on autonomous driving. We also explore the potential benefits and practical applications of these advancements.

9.2 Research Impact and Practical Applications

The findings of this thesis significantly advance the field of autonomous driving by addressing critical challenges in coordination, communication, scalability, security, and real-world applicability. The developed methods enhanced coordination and collaboration between different AVs, which is fundamental to ensuring safety on the roads. These advancements contribute to higher traffic efficiency and system-wide coordination, reducing congestion and improving traffic flow. Furthermore, the ability of AVs to operate cooperatively can lead to optimised fuel consumption, lower emissions, and overall improved environmental sustainability. The robust decision-making capabilities fostered by MARL, through training in diverse scenarios with varying configurations, enhance the adaptability and resilience of AV systems to handle a wide range of dynamic and complex real-world situations effectively.

Improved communication mechanisms, such as the ones presented in Chapter 6, facilitate better AV collaboration in urban environments, directly leading to smoother traffic flow and reduced congestion. This contributes to the development of smarter cities with more efficient and reliable transportation networks. By enabling AVs to navigate complex urban environments safely and efficiently, these ensure seamless navigation through intersections, roundabouts, highway ramp merging, and other urban scenarios. Additionally, these methods enable AVs to safely handle scenarios that are particularly dangerous for human drivers, such as occlusions and unsignalised intersections.

The coordination strategies specifically designed for manoeuvring in the presence of emergency vehicles can significantly enhance emergency response times. By prioritising EMV movement and ensuring that other AVs can safely and efficiently adjust their paths, the proposed methods can reduce delays and improve the overall effectiveness of emergency services. This application has the potential to save lives and improve public safety.

The scalable MARL solutions for fleet management demonstrate the feasibility of managing large-scale autonomous mobility systems. By efficiently managing a fleet of autonomous vehicles, these solutions can optimise ride-sharing services, reduce wait times, and improve vehicle utilisation. The curriculum-based MARL method ensures learning stability and efficiency in dynamic urban environments, making it ideal for real-world deployment in cities with high demand for autonomous transportation services. This paves the way for widespread adoption of autonomous ride-hailing and logistics services, significantly enhancing urban mobility. If implemented effectively and integrated with smart traffic management systems, these services have the potential to reduce traffic congestion and contribute to more efficient urban transportation networks.

Furthermore, the development of adversarial communication defence mechanisms ensures the robustness and security of AV networks against cyber threats, which is crucial for maintaining public trust and the safe operation of autonomous systems. By protecting communication among AVs from adversarial attacks, TrustCommNet helps maintain the integrity and reliability of AV systems. Implementing these methods can help build public trust in autonomous driving

technologies by ensuring that AV systems are resilient against external attacks.

By addressing the reality gap between simulation and real-world applications, this research facilitates the practical deployment of MARL policies. By using domain randomisation during training, the transferability of policies to real-world scenarios is significantly improved. This ensures that AVs can perform reliably in diverse and unpredictable environments, accelerating the deployment of autonomous driving technologies in real-world applications.

Collectively, these contributions have the potential to transform autonomous driving technologies, making them safer, more efficient, and more resilient, thus accelerating their integration into modern transportation systems. In the next section, we discuss the challenges and difficulties associated with these practical applications and explore potential areas for future work.

9.3 Challenges and Future Work

Despite the promising applications of the presented MARL solutions, several challenges must be addressed to facilitate their effective implementation in real-world scenarios. These challenges highlight areas for future research and development:

1. Difficulty in guaranteeing safe policies using MARL: Ensuring consistent safety of MARL policies across all possible scenarios remains a significant challenge. The decentralised and adaptive nature of MARL, combined with the stochastic and dynamic interactions between multiple agents, makes it difficult to predict and guarantee safety outcomes. While MARL can optimise performance, guaranteeing safety in the unpredictable and dynamic real world is complex. Future research should integrate safety constraints directly into the learning process and develop robust verification methods for MARL policies. A promising direction is the extension of Safe Reinforcement Learning (Garcıa and Fernández, 2015) to the multi-agent setting, which presents the additional challenge of coordinating safety across multiple agents.

2. Communication constraints: Packet loss, latency, limited bandwidth, and the inability to communicate effectively can severely impact the performance of MARL-based AV systems.

Real-world communication networks are often unreliable and subject to interference. Developing more resilient communication protocols and exploring redundant paths are crucial for mitigating these issues. In addition, AVs may operate in areas with varying network coverage and quality, leading to inconsistent communication reliability. This inconsistency can result in delays or failures in critical data exchange, affecting the overall performance of AV systems. Future research in this area should incorporate these considerations to better transfer methods to real-world applications.

3. Complexity of real-world scenarios: Real-world driving environments are far more complex than those studied in this thesis. The state and action spaces in real-world scenarios are very large, and the diversity of observations due to different sensors on AVs adds another layer of complexity. Simplified observations used in this research may not capture all necessary real-world dynamics. Thus, scalable algorithms capable of handling this complexity are essential. Techniques such as multi-modal perception, hierarchical reinforcement learning, and transfer learning methods should be explored.

4. Challenges in reward convergence and learning stability: A significant challenge observed in the development of MARL-based solutions, particularly in sim-to-real transfer, is the instability in reward convergence during training. In our experiments, higher levels of noise introduced through domain randomisation led to less stable learning curves, which is not ideal for training. However, this noise improved real-world performance by making policies more robust to unpredictable environments. While smoother convergence without noise may seem preferable, it often results in overfitting to simulation-specific conditions, yielding poor real-world generalisation.

This instability is compounded by several factors beyond added noise. The dynamic and non-stationary nature of multi-agent environments, where each agent's learning affects the others, was identified in Chapter 8 as a key cause of unstable learning. Additionally, the exploration-exploitation trade-off during training further contributes to fluctuations, as agents initially explore suboptimal strategies. Future research should focus on mitigating these challenges by employing adaptive noise injection, refining exploration techniques, and ensuring a balance

between learning stability and policy generalisation to real-world applications.

5. Regulatory and privacy concerns: The implementation of the solutions proposed in this thesis must navigate a complex landscape of regulatory standards and privacy concerns. Regulatory frameworks govern aspects such as vehicle safety standards, communication protocols, and data usage policies to ensure the safety, security, and privacy of users. Current V2X communication standards include the SAE J2735 standard (of Automotive Engineers, 2016), which specifies a common language for V2X messages, and the 3rd Generation Partnership Project (3GPP) standards for Cellular V2X (C-V2X) (3GPP, 2017), which outline the use of cellular networks for V2V communication. Additionally, the IEEE 802.11p standard, part of the Dedicated Short-Range Communications (DSRC) protocol, provides guidelines for wireless access in vehicular environments (IEEE, 2010). A Summary of the V2X technologies and the current standards can be found in Appendix C.

Data sharing between AVs raises significant privacy concerns, as AVs continuously collect and exchange vast amounts of potentially sensitive information. Compliance with privacy laws such as the General Data Protection Regulation (GDPR) in Europe or the CCPA in the United States is essential, requiring strict controls over data collection, storage, and sharing practices. Establishing robust data governance frameworks, including data anonymisation, secure storage, and clear data access protocols, is critical for protecting user privacy and ensuring regulatory compliance.

Obtaining regulatory approval for MARL-based AV systems requires rigorous testing and validation to demonstrate compliance with safety and performance standards, which can be time-consuming and costly. Public perception and acceptance of AV technologies depend on adherence to regulatory standards and the protection of user privacy. Transparent communication about data usage and robust data governance is essential for building public trust and facilitating the widespread adoption of autonomous driving technologies.

6. Inherent challenges of reinforcement learning: RL is known to be sample inefficient, often requiring a large number of interactions with the environment to learn effective policies. This challenge is exacerbated in MARL settings due to the need for coordination among mul-

tiple agents. Additionally, RL in autonomous driving involves sparse rewards, such as those associated with rare events like collisions, making learning slow and unstable. The credit assignment problem, where it is difficult to attribute the outcome of an action to individual agents' decisions, further complicates learning in multi-agent settings. Moreover, effective exploration strategies are necessary to ensure that the learning agents can discover optimal policies in the vast state and action spaces of real-world driving environments. Future research should focus on improving sample efficiency, developing better exploration strategies, and addressing the credit assignment problem in MARL contexts.

7. Bridging the reality gap: While domain randomisation improves the transferability of policies from simulation to the real world, it cannot entirely bridge the reality gap. Future work should focus on increasing simulator fidelity by incorporating more detailed and realistic models of the physical world, including sensor noise, weather conditions, and pedestrian models. Additionally, developing adaptive domain randomisation strategies that can dynamically adjust parameters based on real-world data feedback will be essential.

Methods to quantify and describe reality gaps should be established, potentially through the creation of benchmark datasets that compare simulated and real-world scenarios. Furthermore, implementing continuous learning systems where AVs can learn and adapt from real-world experiences, post-deployment will help refine and enhance the robustness of the policies over time.

In conclusion, while the MARL solutions developed in this thesis hold significant potential for advancing autonomous driving, addressing the practical challenges and closing the gap between research and real-world application are essential for their successful deployment. Continued research, collaboration among industry stakeholders, and regulatory support will be crucial to realise the full potential of these technologies.

Appendix A

Implementation Details

A.1 Coordination in the presence of EMVs (Chapter 5)

A.1.1 Details of the EMMA environment

A scheme of the architecture and logic behind the EMMA environment is presented in Figure A.1. At every step in the simulation: 1) each AV, indexed by i, receives an observation Ω^i from the environment, and the EMV receives Ω^{EMV}; 2) AVs select a high-level action a^i following policy $\pi_{AV}(\Omega^i)$, with the EMV following $a^{EMV} = \pi_{EMV}\left(\Omega^{EMV}\right)$; 3) a non-linear dynamics model (Ercan, 2019) updates the state s^i of the vehicles from a^i; 4) the reward $\mathcal{R}^i$ of each agent is computed according to equation (5.8); 5) rewards are collected across vehicles; 6) if training, the policy is updated. Until a termination condition is reached, a new step ensues. The components in green access or store information in the environment, the components in red refer to the EMV, the ones in black with regular AVs and those in blue are only present when training MAPPO.

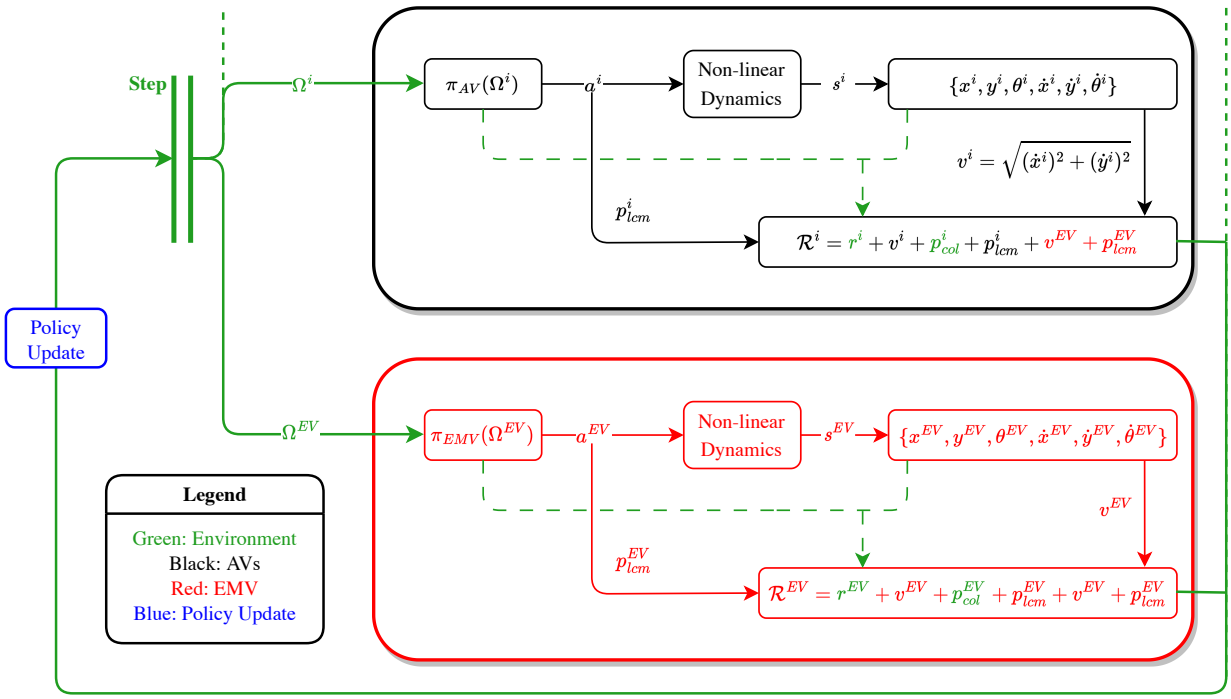


Figure A.1: Architecture of the EMMA environment.

A.1.2 Simulation and training parameters

We used EMMA to train all policies in simulation. We performed a preliminary hyperparameter tunning experiment considering an instance with ten agents (1 EMV and 9 AVs). The combination of the best parameters for the MAPPO are presented in table A.3. The parameters for the risk indices are presented in Table A.2: the acceleration and braking values were selected using the studies on acceleration by (Bokare and Maurya, 2017; Bosetti et al., 2014); the response time ρ was set to match the sampling rate of 10 Hz; the risk propensity parameters β and γ were set to 1, meaning drivers are not assumed to be significantly risk-averse nor risk-prone.

The MARL results were averaged over four different random seeds. The response time was set to the frame rate of the simulations, which was $10Hz$. All experiments were conducted using a 10-core, 3.70Ghz Intel Core i9-10900X CPU and an Nvidia GTX 3090 GPU. Furthermore, the parameters for EMMA are presented in table A.1.

Table A.1: Environment parameters.

Parameter	Values
AV Max. Speed	$20m/s$
AV Min. Speed	$7m/s$
AV Dimensions	$2m \times 4m \times 1.5m$
EMV Max. Speed	$30m/s$
EMV Min. Speed	$7m/s$
Perception radius	$70m$
Road loop length	$400m$

Table A.2: Risk indices parameters

Parameter	Values
Response time ρ	$0.1s$
a_{max}	$2.5m/s^2$
b_{max}	$2.5m/s^2$
b_{min}	$1.0m/s^2$
B	$3.0m/s^2$
a^{lat}_{max}	$1m/s^2$
b^{lat}_{min}	$2.5m/s^2$
B^{lat}	$4.0m/s^2$
β	1
γ	1

A.2 TrustCommNet (Chapter 7)

A.2.1 Classical Environment Details

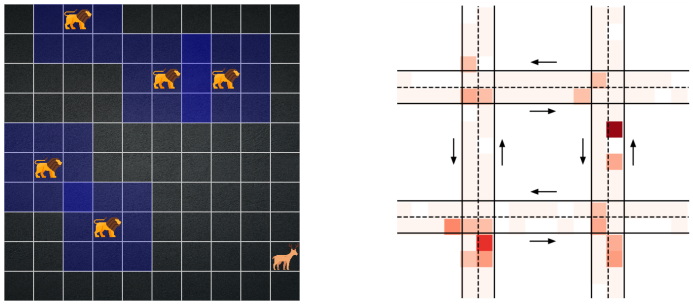
Since our objective is to compare our method against communication-based MARL frameworks, we selected scenarios where these algorithms typically achieve near-perfect performance, i.e. 100% average success rate, and then introduced adversarial attacks to assess their impact on cooperative performance. We use two classical cooperative environments: Predator-Prey and Traffic Junction.

Predator-Prey

We adopt Singh et al.'s Singh et al. (2018)' predator-prey model, where N visually limited predators hunt static prey, moving up, down, left, right, or staying in the same place. Figure A.2a shows a representation of the environment. Predators lose 0.05 points per time step until

Table A.3: MAPPO parameters.

Parameter	Values
N° episodes for training	2000
N° steps per episode	400
Learning Rate	5×10^{-4}
Critic Learning Rate	5×10^{-4}
PPO epochs	15
Actor Network	$MLP(64, 64, 7)$
Critic Network	$MLP(64, 64, 1)$



(a) Predator Prey. (b) Traffic Junction.

Figure A.2: Classical cooperative environments used for the experiments.

they locate the prey. Success is when all predators find the prey within a certain time limit. The environment's difficulty escalates with larger grids and more predators, requiring increased coordination. We opt for a 10×10 grid size with 5 predators and 1 prey.

Traffic Junction

We utilise the implementation by Singh et al. (2018) for our experiments, as shown in Figure A.2b. The environment involves vehicles, each with limited vision, navigating an intersection with a maximum count of N_{max}. Vehicles enter the junction with a probability of p_{arrive} and aim to reach their destinations collision-free, choosing between acceleration and braking. We experiment with two levels of difficulty: easy and medium. In the easy setting, the environment features two one-way roads, a maximum of $N_{max} = 5$ vehicles, and a vehicle arrival probability $p_{arrive} = 0.3$. The medium difficulty introduces two two-way roads, increases N_{max} to 10, and sets p_{arrive} at 0.2.

A.2.2 Adversarial Training Details

The procedure for adversarial training of the baselines and TrustCommNet is described as follows. Initially, all models were pretrained to achieve clear convergence, attaining an average test success rate of over 99%. After this pretraining, the models were exposed to the attacker for the first time. Subsequently, all models were retrained to assess their inherent resilience and ability to counteract the disruptive effects of the malicious messages. During this phase of the experiment, adversarial messages were incorporated from the onset of retraining (step 0) to directly observe the models' adaptive responses.

A.2.3 TrustCommNet Details

The architecture of TrustCommNet is presented in this subsection. The observation encoder employs a linear layer with 256 nodes, while the message encoder uses a 128-node linear layer. The Theory of Mind (ToM) component integrates an LSTM followed by a 128-output linear layer. TrustNet is constructed with three concatenated linear layers paired with ReLU activation functions, transitioning from an input size of $4 \times hidden_size$ to 2. The message reconstructor is designed with two linear layers, and the scheduler incorporates a linear layer coupled with a Gumbel-Softmax layer for decision-making. Table A.4 contains the base hyperparameters of TrustCommNet used in the experiments.

A.2.4 Experiment Details

Our primary evaluation metric is the average success rate (%), or win rate, across all environments and scenarios, indicating the proportion of episodes successfully completed. Additionally, we assess performance using the average team return. To ensure the robustness of training results, we employ 5 independent seeds to generate learning curves, and we validate our experiments across 100 independent test episodes.

The procedure for adversarial training of the baselines and TrustCommNet is described as

Table A.4: Hyper-parameters used to train TrustCommNet

Hyper-parameter	**Value**	**Description**
batch size	500	The size of the batch for training
enc_hid_size	128	The size of hidden units in the message encoder
gat_num_heads	4	The number of heads in the GAT Network
gat_num_heads_out	1	The number of output heads in the GAT Network
gat_hid_size	32	The size of hidden units in the GAT Network
gat_encoder_out_size	32	The size of the encoder's output in the GAT Network
γ	0.9	The discount factor for rewards
normalize_rewards	yes	Whether to normalise rewards or not
learning_rate	0.0001	The learning rate for the optimiser
entropy	0.001	The entropy regularization term
value_coeff	0.015	The coefficient for the value loss
action_scale	1.0	The scaling factor for actions
ϵ	0.3	The epsilon term for adversarial training
ϵ_{steps}	10	The number of steps for adversarial training
η_{trust}	0.5	Trust threshold to filter malicious communication
adversarial_agents	1	The number of adversarial agents
grad_norm_clip	10	The gradient norm clipping value
l2_lambda	0.0001	The lambda for L2 regularization

follows. Initially, all models were pretrained to achieve clear convergence, attaining an average test success rate of over 99%. After this pretraining, the models were exposed to the attacker for the first time. Subsequently, all models were retrained to assess their inherent resilience and ability to counteract the disruptive effects of the malicious messages. During this experiment phase, adversarial messages were incorporated from the onset of retraining (step 0) to directly observe the models' adaptive responses.

All experiments were conducted on a workstation equipped with an Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz processor and an NVIDIA RTX 3090 GPU. The adversarial training times and timesteps for all environments are presented in the table below:

Table A.5: Adversarial Training Times

Environment	Timesteps	Avg. Training Time (minutes)
TJ-5	50,000	35.1
TJ-10	50,000	48.1
PP-5	100,000	58.9

Theoretically, retraining baseline algorithms should provide a defence against malicious agents for two reasons. First, agents may learn to disregard or mitigate adversarial communication. Second, these algorithms are not exclusively dependent on communication for policy and value predictions. Instead, they integrate a combination of encoded messages with their own obser-

vations and hidden states. Thus, they can adapt the network weights to prioritise observations, although this may reduce cooperation effectiveness.

A.2.5 Baseline Details

IC3Net

Ic3Net (Singh et al., 2018) integrates a gating mechanism to manage communication between agents in various environments, including cooperative, semi-cooperative, and competitive settings. The architecture is built upon LSTM units for each agent, enabling the sharing of hidden states through controlled communication, where the decision to communicate is made based on a binary gating action determined by the network.

Ga-Comm

GA-Comm (Liu et al., 2019) employs a two-stage attention mechanism integrated within a graph neural network (GNN) to model the interaction relationships among agents. By abstracting the game into a manageable graph representation where nodes represent agents and edges indicate the interaction relationships, GA-Comm allows each agent to dynamically identify and focus on the most relevant agents for decision-making. This is achieved by first using hard attention to determine whether an interaction between agents exists, followed by soft attention to assess the importance of each interaction. This mechanism enables efficient information exchange and coordination among agents, aiming to enhance learning efficiency and performance in complex environments without the need for pre-defined interaction rules.

TarMAC

TarMAC (Das et al., 2020) enables agents to learn both what messages to send and to whom, within cooperative tasks in partially-observable environments. This architecture introduces a

mechanism for targeted communication among agents, operationalized through a signature-based soft attention mechanism. This allows each agent to broadcast a message accompanied by a key, or signature, which encodes information about the intended recipients, facilitating targeted rather than broadcast communication. The communication strategy is learned end-to-end from task-specific rewards without explicit supervision for communication behaviors. Additionally, TarMAC supports multi-round communication, allowing agents to engage in multiple rounds of message exchange before taking actions, thereby promoting coordinated decision-making.

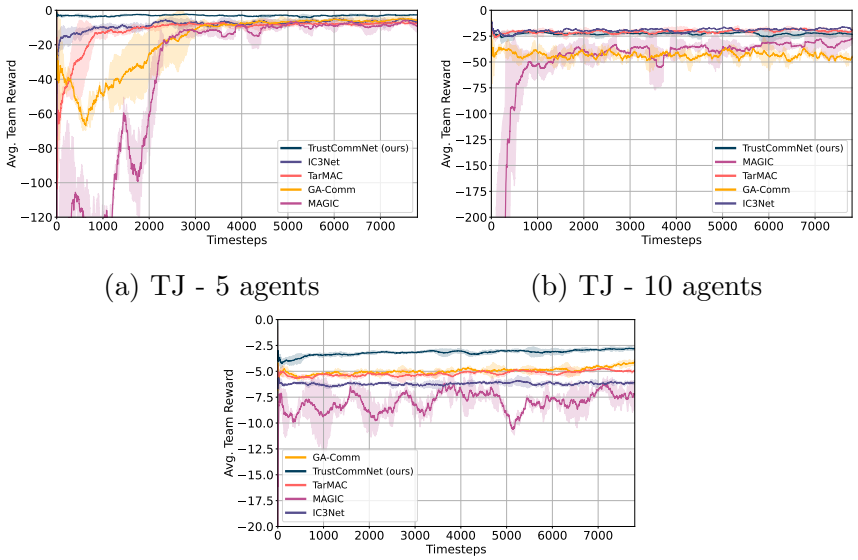
MAGIC

The core architecture of MAGIC (Niu et al., 2021) consists of two primary components: a Scheduler and a Message Processor. The Scheduler is responsible for determining "when" and "whom" each agent should communicate with, leveraging a graph attention mechanism to dynamically generate directed communication graphs that reflect the evolving communication needs among agents. The Message Processor utilizes Graph Attention Networks (GATs) to efficiently integrate and process the received messages based on the communication graphs, facilitating informed decision-making.

A.2.6 Additional Results

Figure A.3 shows the retraining curves for all methods in the Traffic Junction and Predator-Prey environments. TrustCommNet demonstrates rapid recovery in each scenario, reaching high cooperative performance within 8000 steps, unlike baseline algorithms which experience significant disruptions, with some like MAGIC failing to regain prior performance levels.

Figure A.4 showcases TrustCommNet's effectiveness in maintaining robust communication under adversarial attacks. The upper curve shows a significant drop in team success rate at the onset of the attack, highlighting the disruption caused. TrustCommNet quickly identifies the threat, as seen in the rapid trust score decline to zero in the lower heatmaps (agent 0 is the ad-



(a) TJ - 5 agents (b) TJ - 10 agents

(c) PP - 5 agents

Figure A.3: Retraining performance comparison of average team reward.

versary). It then activates its reconstruction mechanisms to neutralise the adversarial impact, quickly recovering previous levels of cooperative performance. The lower heatmaps detail the average trust score η^t for each agent given by TrustNet.

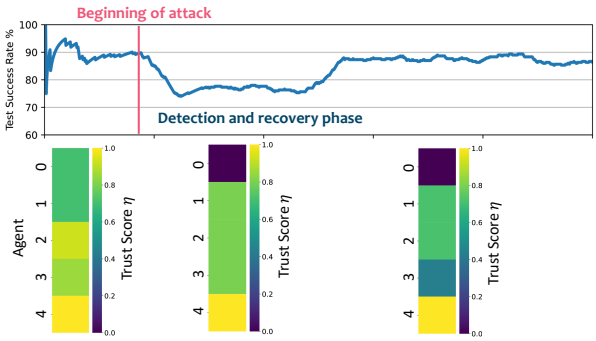


Figure A.4: Response of TrustCommNet to an adversarial attack over time. The top curve displays the test success rate over time. The bottom heatmaps represent the dynamic adjustment of trust scores for individual agents.

A.3 Curriculum-Based DQN (Chapter 8)

A.3.1 PSDQN-CL Implementation details

The practical implementation of the Curriculum-based Parameter Sharing DQN (PSDQN-CL) is described as follows. In the initial stages of the curriculum, the Q-network Q_{θ_0} is trained with the lowest number of agents and the smallest network complexity. The training process in each episode involves selecting actions based on an ϵ-greedy policy, executing these actions, and observing the resulting rewards and the next partial observations $o^{t+1} = (o_1^t, o_2^t, \ldots, o_N^t)$. The transitions are then stored in the replay buffer $\mathcal{D}$, ensuring that the agent's experiences are recorded for future learning.

As the curriculum progresses, the complexity of the tasks increases, which involves adding more agents and evolving the neural network architectures. For each curriculum stage k, a new Q-network Q_{θ_k} is initialised. The agents then continue their training by selecting and executing actions, storing the resulting transitions, and sampling minibatches from the replay buffer to update the Q-network. The standard DQN loss L_{DQN} is computed by calculating the difference between the target Q-value and the current Q-value predicted by the network.

A critical aspect of the PSDQN-CL algorithm is the knowledge transfer mechanism through Q-distillation. When advancing to a new curriculum stage k, the algorithm utilises the previously trained Q-network $Q_{\theta_{k-1}}$ as a teacher to the new network Q_{θ_k}. This involves computing a distillation loss $L_{distill}$ that measures the difference between the Q-values predicted by the current and the previous Q-networks. The total loss L_{total} is then calculated as a combination of the standard DQN loss and the distillation loss, controlled by a hyperparameter λ. This ensures that the new, more complex network retains valuable knowledge from simpler, previously trained networks, leading to more stable and efficient learning.

Throughout the training process, the algorithm dynamically adjusts the number of agents based on the population curriculum scheme. When certain conditions are met, new agents are introduced into the environment. These new agents share the same network weights and replay

buffer with the existing agents, facilitating a consistent and coordinated learning process. Additionally, the demand pattern for customer requests is updated in line with the demand scheme, ensuring that the complexity of the environment scales appropriately with the increasing number of agents.

Finally, the performance of the fully trained Q-network Q_{θ_N} is evaluated on the complete task with the desired number of agents N_d. This evaluation step assesses how well the knowledge distilled and accumulated through the curriculum stages has enabled the agents to handle the full complexity of the environment. By integrating buffer reuse, model reload, and Q-distillation within the curriculum learning framework, PSDQN-CL ensures robust and scalable learning for large-scale autonomous fleet rebalancing.

A.3.2 Hyperparameters

The experimental setup for all subsequent experiments involved using the same DQN architecture. This architecture comprises a neural network with three fully connected hidden layers and an output layer with a single unit for each possible action. Furthermore, all the experiments were conducted using a 4-core, 1.80Ghz Intel Core i7 8550U CPU and an Nvidia GTX 1050 GPU to ensure consistency across experiments. Prior to conducting experiments on the two case studies, we conducted a preliminary set of experiments to determine the best combination of DQN hyperparameters to use in subsequent experiments. The resulting optimal parameters are presented in Table A.6. The hyperparameters of the PSDQN-CL method are presented in Table A.7.

Table A.6: DQN parameters.

Parameter	Values
N° Hidden layers	3
N° Hidden units	512/256/128
Learning rate	5×10^{-4}
Batch size	64
Buffer size	1×10^{6}
ϵ	Linear decay$(1, 0.2, 0.1)$
γ	0.95

Table A.7: Hyperparameters for PSDQN-CL

Hyperparameter	**Description**	**Value**
Replay buffer size	Size of the replay buffer	1×10^6
Minibatch size	Size of the minibatch for updates	32
Learning rate	Learning rate for the Q-network updates	1×10^{-4}
Discount factor (γ)	Discount factor for future rewards	0.99
ϵ-initial	Initial value of ϵ for ϵ-greedy policy	1.0
ϵ-final	Final value of ϵ for ϵ-greedy policy	0.01
ϵ-decay	Decay rate of ϵ	0.99
Number of curriculum stages (N)	Number of stages in the curriculum	5
Network architecture (stage 0)	Architecture of the Q-network at stage 0	1 hidden layer, 128 units
Network architecture (stage 1)	Architecture of the Q-network at stage 1	2 hidden layers, 256 units
Network architecture (stage N)	Architecture of the Q-network at stage N	3 hidden layers, 512 units
Distillation coefficient (λ)	Coefficient for distillation loss	0.1
Training episodes per stage	Number of training episodes per stage	150
Steps per episode	Number of steps per training episode	400

Appendix B

Popular MARL algorithms

B.1 Deep Q-Network

Deep Q-Network (DQN) (Mnih et al., 2015) is a model-free reinforcement learning algorithm that combines Q-learning with deep neural networks. The DQN algorithm seeks to approximate the optimal action-value function $Q^*(s, a)$, which represents the maximum cumulative reward an agent can achieve from a given state s and action a.

The core idea of DQN is to use a neural network to approximate the Q-function. The Q-network takes the state as input and outputs Q-values for all possible actions. The Q-values are updated using the Bellman equation as described in Chapter 2.

DQN introduces two key innovations to stabilise training: experience replay and target networks. Experience replay involves storing the agent's experiences (s, a, r, s') in a replay buffer and sampling mini-batches from this buffer to update the Q-network. This breaks the temporal correlations between experiences and improves data efficiency. The target network, a copy of the Q-network, is used to calculate the target Q-values, and it is updated periodically with the weights of the Q-network to stabilise training.

The DQN algorithm can be summarised as follows:

Algorithm 4 DQN algorithm (Mnih et al., 2015)

Initialise replay buffer $\mathcal{D}$
Initialise action-value function Q with random weights θ
Initialise target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
for episode = 1 to M **do**
  Initialise sequence $s^1 = x^1$ and preprocessed sequence $\phi_1 = \phi(s_1)$
  for t = 1 to T **do**
    With probability ϵ select a random action a^t
    otherwise select $a^t = \arg\max_a Q(\phi(s^t), a; \theta)$
    Execute action a^t in emulator and observe reward r^t and next state s^{t+1}
    Set $s^{t+1} = s^t, a^t, r^t, s^{t+1}$
    Store transition $(\phi(s^t), a^t, r^t, \phi(s^{t+1}))$ in $\mathcal{D}$
    Sample random mini-batch of transitions $(\phi(s^j), a^j, r^j, \phi(s^{j+1}))$ from $\mathcal{D}$
    Set $y^j = \begin{cases} r^j & \text{if episode terminates at step } j+1 \\ r^j + \gamma \max_{a'} \hat{Q}(\phi(s^{j+1}), a'; \theta^-) & \text{otherwise} \end{cases}$
    Perform a gradient descent step on $(y^j - Q(\phi(s^j), a^j; \theta))^2$ with respect to the network parameters θ
    Every C steps reset $\hat{Q} = Q$
  end for
end for

B.2 QMIX

QMIX (Rashid et al., 2018a) is based on the idea of Value Decomposition Networks (VDN), which seek to learn a joint action-value function Q_{tot}, determined by a linear factorisation. Unlike VDN, the QMIX algorithm needs to guarantee that the global $argmax$ is equal to the set of individual $argmaxQ_a$ for each agent:

$$\arg\max_x Q_{tot}(\tau, a) = \begin{pmatrix} \arg\max_{u^1} Q_1(\tau_1, a_1) \\ \vdots \\ \arg\max_{u^n} Q_n(\tau_n, a_n) \end{pmatrix}. \tag{B.1}$$

Where τ is the joint action-observation history and a is the joint action. To guarantee the above, a constraint is added to ensure that the joint action-value function increases monotonically with respect to each individual Q-function:

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \forall a \in A. \tag{B.2}$$

Individual Q_a values are obtained through an agent network, represented as Deep Recurrent Q-

Networks (DRQNs). QMIX uses an additional mixing network (feed-forward) to represent the joint action-value function. During training, the global state s_t is passed to the mixing network, and the DQN update is performed based on the total Q-function Q_{tot}. During execution, each agent can take decentralised actions based exclusively on Q_a.

Algorithm 5 QMIX algorithm (Rashid et al., 2018a)

Initialise θ, the parameters of the mixing network, agent networks and hypernetwork.
Set the learning rate to α and replay buffer $\mathcal{D} =$
$step = 0, \theta' = \theta$
while $step < step_{max}$ **do**
  $t = 0, s = s^0$
  while $s^t \neq$ terminal **and** $t <$ episode limit **do**
    for each agent a **do**
      $\tau^t = \tau^{t-1} \cup \{(o^t, u^{t-1})\}$
      $u^t = \begin{cases} argmax_{u_a^t} Q(\tau_a^t, u_a^t) & \text{with probability } 1-\epsilon \\ randint(1, |U|) & \text{with probability } \epsilon \end{cases}$
    end for
    Get reward r^t and next state s^{t+1}
    $\mathcal{D} = \mathcal{D} \cup \{(s^t), u^t, r^t, s^{t+1}\}$
    $t = t + 1, step = step + 1$
  end while
  if $|\mathcal{D}| > batch\_size$ **then**
    $b \leftarrow$ random batch of episodes from $\mathcal{D}$
    for each timestep t in each episode in batch b **do**
      $Q_{tot} =$ Mixing_network$(Q_1(\tau_t^1, u_t^n);$ Hypernetwork$(s^t; \theta))$
      Calculate target Q_{tot} using Mixing_network with Hypernetwork$(s^t; \theta')$
    end for
    $\Delta Q_{tot} = y_{tot} - Q_{tot}$
    $\Delta\theta = \Delta_\theta (\Delta Q_{tot})^2$
    $\theta = \theta - \alpha \Delta\theta$
  end if
  if update-interval steps have passed **then**
    $\theta' = \theta$
  end if
end while

B.3 MAPPO

MAPPO Algorithm (Yu et al., 2021) extends the Proximal Policy Optimization (PPO) algorithm to multi-agent settings. MAPPO uses a centralised training and decentralised execution framework, where the policy and value networks are trained using joint observations and rewards, but each agent acts based on its local observation.

In the centralised training phase, the policy network $\pi_\theta(a_i|o_i^t)$ and value network $V_\phi(s^t)$ are

trained using *joint observations* s^t and *joint rewards* r^t, where $s^t = (o_1^t, o_2^t, \ldots, o_n^t)$ represents the global state composed of all agents' local observations o_i^t. These joint observations and rewards allow the agents to be trained cooperatively, optimising their actions in the context of the overall system's performance.

However, during the decentralised execution phase, each agent i acts based only on its local observation o_i^t and selects an action a_i^t from its policy network $\pi_\theta(a_i|o_i^t)$, without needing access to the global state. This allows for scalability and flexibility in real-time decision-making, as agents can operate independently in execution while benefiting from the coordination learned during training.

MAPPO leverages the PPO framework, where the policy is updated using the clipped objective function to stabilise training. The policy loss for each agent i is computed as:

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

where $r^t(\theta) = \frac{\pi_\theta(a^t|s^t)}{\pi_{\theta_{\text{old}}}(a^t|s^t)}$ is the ratio between the updated policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$, and $\hat{A}^t$ is the advantage estimate computed using the value function $V_\phi(s^t)$. The advantage function captures how much better the selected action a_i^t is compared to the average action, given the state s^t.

The value loss is used to update the value network V_ϕ by minimising the difference between the predicted value of the state and the actual return. The value loss is computed as:

$$L^V(\phi) = \hat{\mathbb{E}}^t \left[\left(V_\phi(s^t) - \hat{R}^t \right)^2 \right],$$

where $\hat{R}^t$ is the estimated return at time step t, based on the joint reward r^t and future rewards.

During training, each agent's experience is stored in a replay buffer $\mathcal{D}$ as transitions $(s^t, \mathbf{a}^t, r^t, s^{t+1})$, where $\mathbf{a}^t = (a_1^t, a_2^t, \ldots, a_n^t)$ represents the joint action of all agents. A mini-batch of transitions is sampled from $\mathcal{D}$ to compute the policy and value updates. These updates ensure that agents

gradually improve their policies based on the advantage estimates and the value function.

MAPPO's centralised training with decentralised execution allows agents to learn in a coordinated manner but act independently based on their local observations during execution. This makes MAPPO particularly well-suited for applications such as cooperative robotics, autonomous vehicle coordination, and distributed control systems, where joint decision-making is crucial during training but independent operation is essential in deployment.

Algorithm 6 MAPPO Algorithm (Yu et al., 2021)

Initialise policy network π_θ and value network V_ϕ with random weights
Initialise replay buffer $\mathcal{D}$
for episode = 1 to M **do**
  Initialise state s^1
  for t = 1 to T **do**
    for each agent i **do**
      Observe local state o_i^t
      Select action $a_i^t \sim \pi_\theta(a_i|o_i^t)$
    end for
    Execute joint action $\mathbf{a}^t = (a_1^t, a_2^t, \ldots, a_n^t)$
    Observe joint reward r^t and next state s^{t+1}
    Store transition $(s^t, \mathbf{a}^t, r^t, s^{t+1})$ in $\mathcal{D}$
    $s^t \leftarrow s^{t+1}$
  end for
  Sample random mini-batch from $\mathcal{D}$
  Compute advantage estimates $\hat{A}t$ using value network $V\phi$
  for each agent i **do**
    Compute policy loss:
    $L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}^t \left[\min \left(r^t(\theta)\hat{A}^t, \text{clip}(r^t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^t \right) \right]$
    where$r^t(\theta) = \frac{\pi_\theta(a^t|s^t)}{\pi_{\theta_{\text{old}}}(a^t|s^t)}$
    Compute value loss:
    $L^V(\phi) = \hat{\mathbb{E}}^t \left[\left(V_\phi(s^t) - \hat{R}^t \right)^2 \right]$
    Update policy network π_θ via gradient ascent
    Update value network V_ϕ via gradient descent
  end for
end for

Appendix C

V2X Techonology and Standards

C.1 V2X Technology

Vehicle-to-Everything (V2X) technology enables communication between vehicles (V2V), infrastructure (V2I), pedestrians (V2P), and networks (V2N). This technology improves road safety, reduces traffic congestion, and enhances transportation efficiency.

Enabling Technologies: Dedicated Short-Range Communications (DSRC) is a wireless technology operating in the 5.9 GHz band, designed for low-latency, high-reliability communication. Cellular V2X (C-V2X), defined by 3GPP, includes direct communication (PC5) for V2V, V2I, V2P, and network communication (Uu) for V2N, offering better range and scalability than DSRC. Additionally, 5G technology provides ultra-low latency, high data rates, and massive connectivity, crucial for real-time communication and autonomous vehicle integration. Positioning technologies, such as Global Navigation Satellite Systems (GNSS) enhanced by real-time kinematic (RTK) corrections, ensure precise location data for V2X applications.

Integration and Challenges: Integrating V2X technologies into the transportation ecosystem presents challenges, including interoperability, security, privacy, and the need for significant infrastructure investments. Ensuring seamless communication across different systems and robust security measures is critical for the adoption and reliability of V2X communication.

Robust security measures are essential to protect against cyber threats and ensure the reliability of V2X communication.

C.2 Current V2X Standards

This section reviews the key standards regulating V2X data transmission, ensuring interoperability, security, and reliability across different systems and regions.

1. **IEEE 802.11p / ITS-G5**: The IEEE 802.11p standard (IEEE, 2010), also known as Wireless Access in Vehicular Environments (WAVE), is an amendment to the IEEE 802.11 standard tailored for vehicular communication. Operating in the 5.9 GHz band, it supports low-latency V2X communication essential for safety applications. The European counterpart, ITS-G5, defined by the European Telecommunications Standards Institute (ETSI), aligns with IEEE 802.11p to facilitate seamless vehicular communication in Europe.

2. **SAE J2945**: SAE J2945/1 (SAE International, 2016) sets the requirements for On-Board System Requirements for V2V Safety Communications. This standard provides guidelines to ensure that V2V communication systems are interoperable and reliable, focusing on the performance and accuracy of safety messages exchanged between vehicles.

3. **ETSI ITS Standards**: Particularly, the ETSI EN 302 637 is a critical standard within the ETSI ITS framework (ETSI, 2019). It specifies the Cooperative Awareness Basic Service, which is fundamental for V2X communication in Europe. This standard ensures that vehicles can continuously share information about their state, such as position, speed, and direction, with nearby vehicles and infrastructure, facilitating cooperative awareness and collision avoidance.

4. **ISO Standards**: ISO 21217 standard outlines the architecture for the ITS station. This standard provides a comprehensive framework for the components and communication

protocols necessary for ITS, ensuring that different elements of the transportation system can interoperate effectively.

5. **IETF Standards**: The Internet Engineering Task Force (IETF) has developed standards to support IPv6 communication in vehicular environments. RFC 8691, for instance, specifies the use of IPv6 over IEEE 802.11-OCB (Outside the Context of a Basic Service Set), enabling robust and scalable internet connectivity for V2X applications.

6. **WAVE Standards**: The IEEE 1609 (IEEE, 2016) family of standards complements IEEE 802.11p by addressing additional aspects of V2X communication. These include security services (IEEE 1609.2), networking services (IEEE 1609.3), and multi-channel operation (IEEE 1609.4). Together, they form a comprehensive suite for Wireless Access in Vehicular Environments (WAVE), facilitating secure and efficient vehicular communication.

7. **ITU Standards**: The International Telecommunication Union (ITU) provides global standards for V2X communication, focusing on spectrum management and radio interface standards. ITU-R M.2084 (International Telecommunication Union (ITU), 2015) outlines the radio interface standards for public protection and disaster relief operations in the UHF band, which are pertinent for V2X applications in critical communication scenarios.

Bibliography

3GPP (2017). 3rd generation partnership project; technical specification group radio access network; study on lte-based v2x services (release 14).

Adan, F., Feng, Y., Angeloudis, P., Quddus, M., and Ochieng, W. (2023). Constrained multi-agent reinforcement learning policies for cooperative intersection navigation and traffic compliance. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 4079–4085.

Agarwal, A., Kumar, S., and Sycara, K. (2019). Learning Transferable Cooperative Behavior in Multi-Agent Teams. *arXiv:1906.01202 [cs, stat]*. arXiv: 1906.01202.

Ahmed, H. U., Huang, Y., and Lu, P. (2021). A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation. *Smart Cities*, 4(1):314–335.

Alabbasi, A., Ghosh, A., and Aggarwal, V. (2019). DeepPool: Distributed Model-free Algorithm for Ride-sharing using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4714–4727. arXiv: 1903.03882.

Albert, M., Ruch, C., and Frazzoli, E. (2019). Imbalance in Mobility-on-Demand Systems: A Stochastic Model and Distributed Control Approach. *ACM Transactions on Spatial Algorithms and Systems*, 5(2):13:1–13:22.

Albrecht, S. V., Christianos, F., and Schäfer, L. (2024). *Multi-agent reinforcement learning: foundations and modern approaches*. The MIT Press, Cambridge, Massachusetts.

Ali, Y., Haque, M. M., and Zheng, Z. (2022). An Extreme Value Theory approach to estimate crash risk during mandatory lane-changing in a connected environment. *Analytic Methods in Accident Research*, 33:100193.

Alonso, J., Milanés, V., Pérez, J., Onieva, E., González, C., and de Pedro, T. (2011). Autonomous vehicle control systems for safe crossroads. *Transportation Research Part C: Emerging Technologies*, 19(6):1095–1110.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467.

Althoff, M., Stursberg, O., and Buss, M. (2009). Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):299–310.

Aoki, S., Higuchi, T., and Altintas, O. (2020). Cooperative Perception with Deep Reinforcement Learning for Connected Vehicles. arXiv:2004.10927 [cs, eess].

Au, T.-C., Zhang, S., and Stone, P. (2015). Autonomous Intersection Management For Semi-Autonomous Vehicles. In *The Routledge Handbook of Transportation*. Routledge. Num Pages: 17.

Autoware Foundation (2018). Autoware: Open-source software for autonomous driving. Accessed: Sep. 4, 2024.

Bagdadi, O. (2013). Estimation of the severity of safety critical events. *Accident Analysis & Prevention*, 50:167–174.

Baidu (2017). Apollo: An open autonomous driving platform. Accessed: Sep. 4, 2024.

Balaji, B., Bell-Masterson, J., Bilgin, E., Damianou, A., Garcia, P. M., Jain, A., Luo, R., Maggiar, A., Narayanaswamy, B., and Ye, C. (2019a). ORL: Reinforcement Learning Benchmarks for Online Stochastic Optimization Problems. *arXiv:1911.10641 [cs, math]*. arXiv: 1911.10641.

Balaji, B., Mallya, S., Genc, S., Gupta, S., Dirac, L., Khare, V., Roy, G., Sun, T., Tao, Y., Townsend, B., et al. (2019b). Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning. *arXiv preprint arXiv:1911.01562*.

Barceló, J. and Casas, J. (2005). Dynamic network simulation with aimsun. *Simulation approaches in transportation analysis: Recent advances and challenges*, pages 57–98.

Bassani, H. F., Delgado, R. A., Junior, J. N. d. O. L., Medeiros, H. R., Braga, P. H. M., and Tapp, A. (2020). Learning to Play Soccer by Reinforcement and Applying Sim-to-Real to Compete in the Real World. arXiv:2003.11102 [cs, stat].

Bathla, G., Bhadane, K., Singh, R. K., Kumar, R., Aluvalu, R., Krishnamurthi, R., Kumar, A., Thakur, R. N., and Basheer, S. (2022). Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities. *Mobile Information Systems*, 2022(1):7632892. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/7632892.

Belakaria, S., Ammous, M., Sorour, S., and Abdel-Rahim, A. (2017). A Multi-Class Dispatching and Charging Scheme for Autonomous Electric Mobility On-Demand. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–5.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press. Google-Books-ID: rZW4ugAACAAJ.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada. ACM Press.

Bertsimas, D., Jaillet, P., and Martin, S. (2019). Online Vehicle Routing: The Edge of Optimization in Large-Scale Applications. *Operations Research*, 67(1):143–162. Publisher: INFORMS.

Blumenkamp, J. and Prorok, A. (2020). The Emergence of Adversarial Communication in Multi-Agent Reinforcement Learning. *arXiv:2008.02616 [cs]*. arXiv: 2008.02616.

Bokare, P. S. and Maurya, A. K. (2017). Acceleration-deceleration behaviour of various vehicle types. *Transportation research procedia*, 25:4733–4749.

Bosetti, P., Da Lio, M., and Saroldi, A. (2014). On the human control of vehicles: an experimental study of acceleration. *European Transport Research Review*, 6(2):157–170.

Braverman, A., Dai, J. G., Liu, X., and Ying, L. (2019). Empty-Car Routing in Ridesharing Systems. *Operations Research*, 67(5):1437–1452.

Broadhurst, A., Baker, S., and Kanade, T. (2005). Monte carlo road safety reasoning. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 319–324.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.

Brodsky, I. (2018). H3: Uber's Hexagonal Hierarchical Spatial Index.

Brunner, T., Diehl, F., Le, M. T., and Knoll, A. (2019). Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4957–4965. arXiv:1812.09803 [cs, stat].

Buckman, N., Schwarting, W., Karaman, S., and Rus, D. (2021). Semi-Cooperative Control for Autonomous Emergency Vehicles. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7052–7059, Prague, Czech Republic. IEEE.

Cafiso, S., García, A., Cavarra, R., and Rojas, M. A. R. (2011). Crosswalk safety evaluation using a pedestrian risk index as traffic conflict measure. In *3rd International Conference on Road Safety and Simulation*.

Cai, P., Lee, Y., Luo, Y., and Hsu, D. (2020). Summit: A simulator for urban driving in massive mixed traffic. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4023–4029. IEEE.

Candela, E., Doustaly, O., Parada, L., Feng, F., Demiris, Y., and Angeloudis, P. (2023). Risk-aware controller for autonomous vehicles using model-based collision prediction and reinforcement learning. *Artificial Intelligence*, 320:103923.

Candela, E., Parada, L., Marques, L., Georgescu, T.-A., Demiris, Y., and Angeloudis, P. (2022). Transferring Multi-Agent Reinforcement Learning Policies for Autonomous Driving using Sim-to-Real. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8814–8820. ISSN: 2153-0866.

Cao, Y., Xu, D., Weng, X., Mao, Z., Anandkumar, A., Xiao, C., and Pavone, M. (2023). Robust Trajectory Prediction against Adversarial Attacks. In *Proceedings of The 6th Conference on Robot Learning*, pages 128–137. PMLR. ISSN: 2640-3498.

Chen, C., Yao, F., Mo, D., Zhu, J., and Chen, X. M. (2021). Spatial-temporal pricing for ride-sourcing platform with reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 130:103272.

Chen, D., Li, Z., Hajidavalloo, M., Chen, K., Wang, Y., Jiang, L., and Wang, Y. (2022). Deep Multi-agent Reinforcement Learning for Highway On-Ramp Merging in Mixed Traffic. *arXiv:2105.05701 [cs, eess]*. arXiv: 2105.05701.

Chen, Q., Tang, S., Yang, Q., and Fu, S. (2019). Cooper: Cooperative Perception for Connected Autonomous Vehicles based on 3D Point Clouds. arXiv:1905.05265 [cs].

Chevalier, M., Golemo, F., Cao, Y., Mehta, B., and Paull, L. (2018). Duckietown environments for openai gym. `https://github.com/duckietown/gym-duckietown`.

Chia, W. M. D., Keoh, S. L., Goh, C., and Johnson, C. (2022). Risk Assessment Methodologies for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16923–16939. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Chowdhury, A. (2016). Priority based and secured traffic management system for emergency vehicle using IoT. In *2016 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6, Agadir, Morocco. IEEE.

Chu, T., Wang, J., Codeca, L., and Li, Z. (2020). Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095.

Claussmann, L., Revilloud, M., Gruyer, D., and Glaser, S. (2020). A Review of Motion Planning for Highway Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1826–1848. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Codevilla, F., Santana, E., López, A. M., and Gaidon, A. (2019). Exploring the Limitations of Behavior Cloning for Autonomous Driving. arXiv:1904.08980 [cs].

Cordeau, J.-F. (2006). A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Operations Research*, 54(3):573–586. Publisher: INFORMS.

Cubillos, C., Guidi-Polanco, F., and Demartini, C. (2004). *Multi-agent infrastructure for distributed planning of demand-responsive passenger transportation service*, volume 2. Journal Abbreviation: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics Pages: 2017 vol.2 Publication Title: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics.

Cui, J., Qiu, H., Chen, D., Stone, P., and Zhu, Y. (2022). COOPERNAUT: End-to-End Driving with Cooperative Perception for Networked Vehicles. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17231–17241. arXiv:2205.02222 [cs].

Cunto, F. and Saccomanno, F. F. (2008). Calibration and validation of simulated vehicle safety performance at signalized intersections. *Accident Analysis & Prevention*, 40(3):1171–1179.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., and Pineau, J. (2020). TarMAC: Targeted Multi-Agent Communication. arXiv:1810.11187 [cs, stat].

de Lima, O., Shah, H., Chu, T.-S., and Fogelson, B. (2020). Efficient Ridesharing Dispatch Using Multi-Agent Reinforcement Learning. *arXiv:2006.10897 [cs, stat]*. arXiv: 2006.10897.

DeepDrive Team (2019). Deepdrive autonomous driving simulator. Accessed: Sep. 4, 2024.

Deng, Y., Chen, H., Shao, S., Tang, J., Pi, J., and Gupta, A. (2020). Multi-Objective

Vehicle Rebalancing for Ridehailing System using a Reinforcement Learning Approach. *arXiv:2007.06801 [cs, eess]*. arXiv: 2007.06801.

Dinneweth, J., Boubezoul, A., Mandiau, R., and Espié, S. (2022). Multi-agent reinforcement learning for autonomous vehicles: a survey. *Autonomous Intelligent Systems*, 2(1):27.

Dong, J., Chen, S., Ha, P. Y. J., Li, Y., and Labi, S. (2020). A DRL-based Multiagent Cooperative Control Framework for CAV Networks: a Graphic Convolution Q Network. *arXiv:2010.05437 [cs, eess]*. arXiv: 2010.05437.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An Open Urban Driving Simulator. arXiv:1711.03938 [cs].

Dresner, K. and Stone, P. (2006). Human-Usable and Emergency Vehicle–Aware Control Policies for Autonomous Intersection Management. page 9.

Dupuis, M., Strobl, M., and Grezlikowski, H. (2010). OpenDRIVE 2010 and Beyond – Status and Future of the de facto Standard for the Description of Road Networks. page 12.

Eggert, J. (2014). Predictive risk estimation for intelligent adas functions. In *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*.

Eidehall, A. (2011). Multi-target threat assessment for automotive applications. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 433–438.

Ercan, S. (2019). Modelling strategies for a mobile robot. Master's thesis, Swiss Federal Institute of Technology Zurich.

Espadinha, J., Lebedev, I., Lukic, L., and Bernardino, A. (2021). LiDAR Data Noise Models and Methodology for Sim-to-Real Domain Generalization and Adaptation in Autonomous Driving Perception. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 797–803.

ETSI (2014). Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.

ETSI (2019). ETSI EN 302 637-2 v1.4.1: Intelligent transport systems (ITS); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service.

Eysenbach, B., Gu, S., Ibarz, J., and Levine, S. (2017). Leave no trace: Learning to reset for safe and autonomous reinforcement learning. *arXiv preprint arXiv:1711.06782*.

Farley, B. and Clark, W. (1954). Simulation of self-organizing systems by digital computer. *Transactions of the IRE Professional Group on Information Theory*, 4(4):76–84. Conference Name: Transactions of the IRE Professional Group on Information Theory.

Feth, P., Akram, M. N., Schuster, R., and Wasenmüller, O. (2018). Dynamic risk assessment for vehicles of higher automation levels by deep learning. In *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11094 of *Lecture Notes in Computer Science*, pages 535–547. Springer, Cham, Switzerland.

Figura, M., Kosaraju, K. C., and Gupta, V. (2021). Adversarial attacks in consensus-based multi-agent reinforcement learning. arXiv:2103.06967 [cs, eess, stat].

Fitch, G., Lee, S., Klauer, S., Hankey, J., Sudweeks, J., and Dingus, T. (2009). Analysis of lane-change crashes and near-crashes. *US Department of Transportation, National Highway Traffic Safety Administration*.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. (2017). Counterfactual Multi-Agent Policy Gradients. *arXiv:1705.08926 [cs]*. arXiv: 1705.08926.

Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P., and Whiteson, S. (2018). Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning. *arXiv:1702.08887 [cs]*. arXiv: 1702.08887.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. (2016). Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *arXiv:1605.06676 [cs]*. arXiv: 1605.06676.

Fogelson, B., Shah, H., de Lima, O., and Chu, T.-S. (2020). Efficient Ridesharing Dispatch Using Multi-Agent Reinforcement Learning. page 12.

Formosa, N., Quddus, M., Ison, S., and Timmis, A. (2022). A new modeling approach for predicting vehicle-based safety threats. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):18175–18185.

Fung, H. L., Darvariu, V.-A., Hailes, S., and Musolesi, M. (2022). Trust-based Consensus in Multi-Agent Reinforcement Learning Systems. arXiv:2205.12880 [cs].

Garcia, J. and Fernández, F. (2012). Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564.

Garcıa, J. and Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480.

Garrido, F. and Resende, P. (2022). Review of decision-making and planning approaches in automated driving. *IEEE Access*, 10:100348–100366.

Gayle, D. (2019). Number of road deaths involving police vehicles at 13-year high. *The Guardian*.

Geibel, P. and Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108.

Gipps, P. G. (1986). A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological*, 20(5):403–414.

González, C. L., Pulido, J. J., Alberola, J. M., Julian, V., and Niño, L. F. (2021). Autonomous Distributed Intersection Management for Emergency Vehicles at Intersections. In De La Prieta, F., El Bolock, A., Durães, D., Carneiro, J., Lopes, F., and Julian, V., editors, *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection*, pages 261–269, Cham. Springer International Publishing.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [cs, stat].

Google (2017). Draco: 3d data compression. Open-source library for compressing and decompressing 3D geometric meshes and point clouds.

Goslin, M. and Mine, M. (2004). The Panda3D Graphics Engine. *Computer*, 37:112–114.

Gueriau, M., Cugurullo, F., Acheampong, R. A., and Dusparic, I. (2020). Shared Autonomous Mobility on Demand: A Learning-Based Approach and Its Performance in the Presence of Traffic Congestion. *IEEE Intelligent Transportation Systems Magazine*, 12(4):208–218. Conference Name: IEEE Intelligent Transportation Systems Magazine.

Gueriau, M. and Dusparic, I. (2018). SAMoD: Shared Autonomous Mobility-on-Demand using Decentralized Reinforcement Learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1558–1563, Maui, HI. IEEE.

Gupta, A., Yu, J., Zhao, T. Z., Kumar, V., Rovinsky, A., Xu, K., Devlin, T., and Levine, S. (2021). Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6664–6671. IEEE.

Hafner, M. R., Cunningham, D., Caminiti, L., and Del Vecchio, D. (2013). Cooperative Collision Avoidance at Intersections: Algorithms and Experiments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1162–1175. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Hayward, J. C. (1972). Near-miss determination through use of a scale of danger. In *51st Annual Meeting of the Highway Research Board*, number 384 in Highway Research Record, pages 24–34, Washington, D.C. Highway Research Board. Sponsored by Committee on Effectiveness of Operational Measures.

Hillenbrand, J., Spieker, A. M., and Kroschel, K. (2006). A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):528–540.

Holler, J., Vuorio, R., Qin, Z., Tang, X., Jiao, Y., Jin, T., Singh, S., Wang, C., and Ye, J.

(2019). Deep Reinforcement Learning for Multi-Driver Vehicle Dispatching and Repositioning Problem. *arXiv:1911.11260 [cs, stat]*. arXiv: 1911.11260.

Horn, M. E. (2002). Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, 10(1):35–63.

Hu, T., Hu, Z., Lu, Z., and Wen, X. (2023). Dynamic traffic signal control using mean field multi-agent reinforcement learning in large scale road-networks. *IET Intelligent Transport Systems*.

Humayun, M., Almufareh, M. F., and Jhanjhi, N. Z. (2022). Autonomous Traffic System for Emergency Vehicles. *Electronics*, 11(4):510. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Hörl, S., Ruch, C., Becker, F., Frazzoli, E., and Axhausen, K. (2019). Fleet operational policies for automated mobility: A simulation assessment for Zurich. *Transportation Research Part C: Emerging Technologies*, 102:20–31.

IEEE (2010). IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications amendment 6: Wireless access in vehicular environments.

IEEE (2016). IEEE 1609 family of standards for wireless access in vehicular environments (WAVE).

International Telecommunication Union (ITU) (2015). ITU-R M.2084: Radio interface standards for use by public protection and disaster relief operations in some parts of the uhf band in accordance with resolution 646 (WRC-03).

IPG Automotive (2023). CarMaker Vehicle Simulation Software. Accessed: Sep. 4, 2024.

Isele, D., Rahimi, R., Cosgun, A., Subramanian, K., and Fujimura, K. (2018). Navigating Occluded Intersections with Autonomous Vehicles using Deep Reinforcement Learning. arXiv:1705.01196 [cs].

Isidore, C. and Valdes-Dapena, P. (2021). Tesla is under investigation because its cars keep hitting emergency vehicles.

ITS Deployment Evaluation (2023). Advanced driver assistance systems (adas): Cooperative adaptive cruise control (cacc) — its deployment evaluation.

Jansson, J. (2005). *Collision Avoidance Theory with Application to Automotive Collision Mitigation*. PhD thesis, Linkoping University.

Jiang, J. and Lu, Z. (2018). Learning Attentional Communication for Multi-Agent Cooperation. arXiv:1805.07733 [cs].

Jiao, Y., Tang, X., Qin, Z. T., Li, S., Zhang, F., Zhu, H., and Ye, J. (2021). Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 130:103289.

Jindal, I., Qin, Z., Chen, X., Nokleby, M., and Ye, J. (2018). Optimizing Taxi Carpool Policies via Reinforcement Learning and Spatio-Temporal Mining. *arXiv:1811.04345 [cs, stat]*. arXiv: 1811.04345.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101:99–134.

Kalweit, G., Huegle, M., Werling, M., and Boedecker, J. (2020). Deep Constrained Q-learning. arXiv:2003.09398 [cs, stat].

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. arXiv:2004.04450 [cs].

Karmakar, G., Chowdhury, A., Kamruzzaman, J., and Gondal, I. (2021). A Smart Priority-Based Traffic Control System for Emergency Vehicles. *IEEE Sensors Journal*, 21(14):15849–15858. Conference Name: IEEE Sensors Journal.

Kaspar, M., Osorio, J. D. M., and Bock, J. (2020). Sim2Real Transfer for Reinforcement Learning without Dynamics Randomization. arXiv:2002.11635 [cs].

Kazari, K., Shereen, E., and Dan, G. (2023). Decentralized Anomaly Detection in Cooperative Multi-Agent Reinforcement Learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 162–170, Macau, SAR China. International Joint Conferences on Artificial Intelligence Organization.

Kenney, J. B. (2011). Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182. Conference Name: Proceedings of the IEEE.

Kesting, A., Treiber, M., and Helbing, D. (2007). General lane-changing model mobil for car-following models. *Transportation Research Record*, 1999(1):86–94.

Kim, K., Kim, J. S., Jeong, S., Park, J.-H., and Kim, H. K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers & Security*, 103:102150.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., and Pérez, P. (2022). Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3.

Koopman, P. and Wagner, M. (2016). Challenges in Autonomous Vehicle Testing and Validation. *SAE International Journal of Transportation Safety*, 4(1):15–24. Publisher: SAE International.

Krasowski, H., Wang, X., and Althoff, M. (2020). Safe Reinforcement Learning for Autonomous Lane Changing Using Set-Based Prediction. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7.

Kullman, N. D., Cousineau, M., Goodson, J. C., and Mendoza, J. E. (2022). Dynamic Ride-Hailing with Electric Vehicles. *Transportation Science*, 56(3):775–794. Publisher: INFORMS.

Kurakin, A., Goodfellow, I., and Bengio, S. (2017). Adversarial Machine Learning at Scale. arXiv:1611.01236 [cs, stat].

Lamport, L., Shostak, R., and Pease, M. (2019). The Byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. Association for Computing Machinery, New York, NY, USA.

Laurent, G. J., Matignon, L., and Le Fort-Piat, N. (2011). The world of independent learners is not markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15(1):55–64.

Lazar, H., Rhoulami, K., and Rahmani, D. (2016). A Review Analysis of Optimal Velocity Models. *Periodica Polytechnica Transportation Engineering*, 44(2):123–131. Number: 2.

Lecerf, U., Yemdji-Tchassi, C., and Michiardi, P. (2022). Safer Autonomous Driving in a Stochastic, Partially-Observable Environment by Hierarchical Contingency Planning. Technical Report arXiv:2204.06509, arXiv. arXiv:2204.06509 [cs] type: article.

Leurent, E. (2018). An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`.

Li, M., Zhiwei, Qin, Jiao, Y., Yang, Y., Gong, Z., Wang, J., Wang, C., Wu, G., and Ye, J. (2019). Efficient Ridesharing Order Dispatching with Mean Field Multi-Agent Reinforcement Learning. *arXiv:1901.11454 [cs]*. arXiv: 1901.11454.

Li, N., Oyler, D., Zhang, M., Yildiz, Y., Kolmanovsky, I., and Girard, A. (2016). Game-Theoretic Modeling of Driver and Vehicle Interactions for Verification and Validation of Autonomous Vehicle Control Systems. *arXiv:1608.08589 [cs]*. arXiv: 1608.08589.

Li, Q., Peng, Z., Feng, L., Zhang, Q., Xue, Z., and Zhou, B. (2022). MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning. arXiv:2109.12674 [cs].

Li, S., Guo, J., Xiu, J., Feng, P., Yu, X., Liu, A., Wu, W., and Liu, X. (2023a). Attacking Cooperative Multi-Agent Reinforcement Learning by Adversarial Minority Influence. arXiv:2302.03322 [cs].

Li, Y., Fang, Q., Bai, J., Chen, S., Juefei-Xu, F., and Feng, C. (2023b). Among Us: Adversarially Robust Collaborative Perception by Consensus. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 186–195, Paris, France. IEEE.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv:1509.02971 [cs, stat] version: 1.

Lin, K., Zhao, R., Xu, Z., and Zhou, J. (2018). Efficient Large-Scale Fleet Management via Multi-Agent Deep Reinforcement Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1774–1783, London United Kingdom. ACM.

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. (2019). Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. arXiv:1703.06748 [cs, stat].

Liu, Y., Wang, W., Hu, Y., Hao, J., Chen, X., and Gao, Y. (2019). Multi-Agent Game Abstraction via Graph Attention Neural Network. arXiv:1911.10715 [cs].

Liu, Y.-C., Tian, J., Glaser, N., and Kira, Z. (2020a). When2com: Multi-Agent Perception via Communication Graph Grouping. arXiv:2006.00176 [cs].

Liu, Y.-C., Tian, J., Ma, C.-Y., Glaser, N., Kuo, C.-W., and Kira, Z. (2020b). Who2com: Collaborative Perception via Learnable Handshake Communication. arXiv:2003.09575 [cs].

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wiessner, E. (2018). Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2020). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv:1706.02275 [cs].*

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

Lu, Q. and Kim, K.-D. (2017). A Genetic Algorithm Approach for Expedited Crossing of Emergency Vehicles in Connected and Autonomous Intersection Traffic. *Journal of Advanced Transportation*, 2017:1–14.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2019). Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [cs, stat].

Mao, C., Liu, Y., and Shen, Z.-J. M. (2020). Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 115:102626.

Mariani, S., Cabri, G., and Zambonelli, F. (2022). Coordination of Autonomous Vehicles: Taxonomy and Survey. *ACM Computing Surveys*, 54(1):1–33. arXiv:2001.02443 [cs].

Marshall, A. (2022). An Autonomous Car Blocked a Fire Truck Responding to an Emergency. *Wired*. Section: tags.

Matas, J., James, S., and Davison, A. J. (2018). Sim-to-Real Reinforcement Learning for Deformable Object Manipulation. arXiv:1806.07851 [cs].

McGee, P. (2021). Robotaxis: have Google and Amazon backed the wrong technology? *Financial Times*.

Mechanical Simulation Corporation (2023). CarSim Vehicle Simulation Software. Accessed: Sep. 4, 2024.

Miao, F., Han, S., Lin, S., Stankovic, J. A., Zhang, D., Munir, S., Huang, H., He, T., and Pappas, G. J. (2016). Taxi Dispatch With Real-Time Sensing Data in Metropolitan Areas: A Receding Horizon Control Approach. *IEEE Transactions on Automation Science and Engineering*, 13(2):463–478.

Michel, O. (2004). Cyberbotics ltd. webots™: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5.

Miculescu, D. and Karaman, S. (2016). Polling-systems-based Autonomous Vehicle Coordination in Traffic Intersections with No Traffic Signals. *arXiv:1607.07896 [cs, math]*. arXiv: 1607.07896.

Minderhoud, M. M. and Bovy, P. H. L. (2001). Extended time-to-collision measures for road traffic safety assessment. *Accident Analysis & Prevention*, 33(1):89–97.

Minsky, M. L. (1954). *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. Thesis, Princeton University.

Mitchell, R., Blumenkamp, J., and Prorok, A. (2020). Gaussian Process Based Message Filtering for Robust Multi-Agent Cooperation in the Presence of Adversarial Communication. arXiv:2012.00508 [cs].

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. *arXiv:1602.01783 [cs]*. arXiv: 1602.01783.

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(7540):529–533. Publisher: Nature Publishing Group.

Mo, K., Tang, W., Li, J., and Yuan, X. (2023). Attacking Deep Reinforcement Learning With Decoupled Adversarial Policy. *IEEE Transactions on Dependable and Secure Computing*, 20(1):758–768. Conference Name: IEEE Transactions on Dependable and Secure Computing.

Muzahid, A. J. M., Kamarulzaman, S. F., Rahman, M. A., Murad, S. A., Kamal, M. A. S., and Alenezi, A. H. (2023). Multiple vehicle cooperation and collision avoidance in automated vehicles: survey and an AI-enabled conceptual framework. *Scientific Reports*, 13(1):603. Publisher: Nature Publishing Group.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. (2017). Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. arXiv:1708.02596 [cs].

Nageshrao, S., Tseng, H. E., and Filev, D. (2019). Autonomous Highway Driving using Deep Reinforcement Learning. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2326–2331. ISSN: 2577-1655.

Namazi, E., Li, J., and Lu, C. (2019). Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic Literature Review. *IEEE Access*, 7:91946–91965. Conference Name: IEEE Access.

Nellore, K. and Hancke, G. (2016). Traffic Management for Emergency Vehicle Priority Based on Visual Sensing. *Sensors*, 16(11):1892.

New York City Taxi and Limousine Commission (2021). TLC Trip Record Data - TLC.

Ngai, D. C. K. and Yung, N. H. C. (2011). A Multiple-Goal Reinforcement Learning Method for Complex Vehicle Overtaking Maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):509–522. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Ngu, E., Parada, L., Escribano Macias, J. J., and Angeloudis, P. (2022). Decentralised Multi-Agent Reinforcement Learning Approach for the Same-Day Delivery Problem. *Transportation Research Record: Journal of the Transportation Research Board*, 2676(11):385–395.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. arXiv:1412.1897 [cs].

Niu, Y., Paleja, R., and Gombolay, M. (2021). Multi-Agent Graph-Attention Communication and Teaming.

NSC (2020). Emergency vehicles – injury facts.

Nyberg, T., Pek, C., Dal Col, L., Norén, C., and Tumova, J. (2021). Risk-aware motion planning for autonomous vehicles with safety specifications. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 1016–1023. IEEE.

Oda, T. and Joe-Wong, C. (2018). MOVI: A Model-Free Approach to Dynamic Fleet Management. *arXiv:1804.04758 [cs, stat]*. arXiv: 1804.04758.

Oda, T. and Tachibana, Y. (2018). Distributed Fleet Control with Maximum Entropy Deep Reinforcement Learning.

of Automotive Engineers, S. (2016). Dedicated short range communications (dsrc) message set dictionary.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. (2017). Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability. *arXiv:1703.06182 [cs]*. arXiv: 1703.06182.

OptiTrack (2022). Optitrack: Motion capture systems. Accessed: 2024-06-25.

OroojlooyJadid, A. and Hajinezhad, D. (2020). A Review of Cooperative Multi-Agent Deep Reinforcement Learning. *arXiv:1908.03963 [cs, math, stat]*. arXiv: 1908.03963.

Palade, V. and Deo, A. (2022). Artificial intelligence in cars: How close yet far are we from fully autonomous vehicles? *International Journal on Artificial Intelligence Tools*, 31(3):2241005.

Palanisamy, P. (2019a). Multi-Agent Connected Autonomous Driving using Deep Reinforcement Learning. page 16.

Palanisamy, P. (2019b). Multi-agent connected autonomous driving using deep reinforcement learning.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical Black-Box Attacks against Machine Learning. arXiv:1602.02697 [cs].

Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. (2019). Dealing with Non-Stationarity in Multi-Agent Deep Reinforcement Learning. arXiv:1906.04737 [cs, stat].

Parada, L. and Angeloudis, P. (2024). Securing autonomous vehicle networks: Strategic defence against communication attacks. In *RSS 2024 Workshop for Autonomous Vehicles Across Scale (AVAS)*, Delft, Netherlands.

Parada, L., Candela, E., Marques, L., and Angeloudis, P. (2023a). Safe and efficient manoeuvring for emergency vehicles in autonomous traffic using multi-agent proximal policy optimisation. *Transportmetrica A: Transport Science*, 0(0):1–29. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/23249935.2023.2246586.

Parada, L., Escribano, J., Feng, Y., Karamanis, R., and Angeloudis, P. (2023b). Scaling Deep Reinforcement Learning for Autonomous Mobility on Demand Systems: A Curriculum Learning Approach.

Parada, L., Tian, H., and Angeloudis, P. (2023c). Assessing the Robustness of Multi-Agent Deep Reinforcement Learning for Collaborative Navigation under Adversarial V2V Attacks. In *10th International Symposium on Transportation Data & Modelling (ISTDM2023)*.

Parada, L., Tian, H., Escribano, J., and Angeloudis, P. (2023d). An End-to-End Collaborative Learning Approach for Connected Autonomous Vehicles in Occluded Scenarios. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5548–5554. ISSN: 2153-0017.

Parada, L., Tian, H., Escribano, J., and Angeloudis, P. (2023e). A V2V-based Collaborative Navigation Framework for Connected Autonomous Vehicles in Occluded Scenarios. In *ICRA Workshop on Scalable Autonomous Driving (2023)*.

Parada, L., Yu, K., and Angeloudis, P. (2024). Cooperative Autonomous Driving via Intention-Sharing Multi-Agent Reinforcement Learning.

Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., and Chowdhary, G. (2017). Robust Deep Reinforcement Learning with Adversarial Attacks. arXiv:1712.03632 [cs].

Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y. F., Choi, C., Dusek, J., Fang, Y., et al. (2017). Duckietown: an open, inexpensive and flexible platform

for autonomy education and research. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1497–1504. IEEE.

Pavone, M. (2015). Autonomous Mobility-on-Demand Systems for Future Urban Mobility. In Maurer, M., Gerdes, J. C., Lenz, B., and Winner, H., editors, *Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte*, pages 399–416. Springer, Berlin, Heidelberg.

Pavone, M., Smith, S., Frazzoli, E., and Rus, D. (2012). Robotic load balancing for mobility-on-demand systems. *International Journal of Robotic Research - IJRR*, 31:839–854.

Peng, B., Wang, J., and Zhang, Z. (2020). A Deep Reinforcement Learning Algorithm Using Dynamic Attention Model for Vehicle Routing Problems. *arXiv:2002.03282 [cs, stat]*. arXiv: 2002.03282.

Psaraftis, H. N., Wen, M., and Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31.

PTV Group (1993). PTV Vissim: Multimodal Traffic Simulation Software. Accessed: Sep. 4, 2024.

Punzo, V. and Tripodi, A. (2007). Steady-State Solutions and Multiclass Calibration of Gipps Microscopic Traffic Flow Model. *Transportation Research Record*, 1999:104–114.

Puterman, M. L. (1994). Markov decision processes: Discrete stochastic dynamic programming.

Qin, Z. T., Zhu, H., and Ye, J. (2022). Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144:103852.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S. M. A., and Botvinick, M. (2018). Machine Theory of Mind. arXiv:1802.07740 [cs].

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. (2018a). QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv:1803.11485 [cs, stat]*.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. (2018b). Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR.

Rawashdeh, Z. Y. and Wang, Z. (2018). Collaborative Automated Driving: A Machine Learning-based Method to Enhance the Accuracy of Shared Information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3961–3966. ISSN: 2153-0017.

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., Agafonov, E., Kim, T. H., Sterner, E., Ushiroda, K., Reyes, M., Zelenkovsky, D., and Kim, S. (2020). Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. (2016). Policy Distillation. arXiv:1511.06295 [cs].

SAE (2019). J3016C: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International.

SAE International (2016). SAE J2945/1: On-board system requirements for V2V safety communications.

Schwarting, W., Alonso-Mora, J., and Rus, D. (2018). Planning and Decision-Making for Autonomous Vehicles. page 26.

Sclar, M., Neubig, G., and Bisk, Y. (2022). Symmetric Machine Theory of Mind. In *Proceedings of the 39th International Conference on Machine Learning*, pages 19450–19466. PMLR. ISSN: 2640-3498.

Scott, H., Niccolini, L., Stetson, C., Goldbeck, N., Souflas, I., Songur, N., Ahrabian, A., Candela, E., and Angeloudis, P. (2023). Scenario Gym: A Scenario-Centric Lightweight Sim-

ulator. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5216–5222. ISSN: 2153-0017.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In Hutter, M. and Siegwart, R., editors, *Field and Service Robotics*, pages 621–635, Cham. Springer International Publishing.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv:1610.03295 [cs, stat]*. arXiv: 1610.03295.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2017). On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*.

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton.

Shawky, M. (2020). Factors affecting lane change crashes. *IATSS research*, 44(2):155–161.

Shelke, M., Malhotra, A., and Mahalle, P. N. (2019). Fuzzy priority based intelligent traffic congestion control and emergency vehicle management using congestion-aware routing algorithm. *Journal of Ambient Intelligence and Humanized Computing*.

Shi, J., Gao, Y., Wang, W., Yu, N., and Ioannou, P. A. (2020). Operating Electric Vehicle Fleet for Ride-Hailing Services With Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4822–4834. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Shou, Z. and Di, X. (2020). Reward design for driver repositioning using multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 119:102738.

Shu, T., Bhandwaldar, A., Gan, C., Smith, K. A., Liu, S., Gutfreund, D., Spelke, E., Tenenbaum, J. B., and Ullman, T. D. (2021). AGENT: A Benchmark for Core Psychological Reasoning. arXiv:2102.12321 [cs].

Shu, T. and Tian, Y. (2019). M$ˆ3$RL: Mind-aware Multi-agent Management Reinforcement Learning. arXiv:1810.00147 [cs, stat].

Shuo Ma, Yu Zheng, and Wolfson, O. (2013). T-share: A large-scale dynamic taxi ridesharing service. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 410–421, Brisbane, QLD. IEEE.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2017a). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arXiv:1712.01815 [cs].

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017b). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359. Publisher: Nature Publishing Group.

Simoudis, E. (2020). *Transportation Transformation: How Autonomous Mobility Will Fuel New Value Chains*. S.l.

Singh, A., Jain, T., and Sukhbaatar, S. (2018). Learning when to Communicate at Scale in Multiagent Cooperative and Competitive Tasks. arXiv:1812.09755 [cs, stat].

So, J., Xie, A., Jung, S., Edlund, J., Thakker, R., Agha-mohammadi, A., Abbeel, P., and James, S. (2022). Sim-to-Real via Sim-to-Seg: End-to-end Off-road Autonomous Driving Without Real Data. arXiv:2210.14721 [cs].

So, J. J., Kang, J., Park, S., Park, I., and Lee, J. (2020). Automated Emergency Vehicle Control Strategy Based on Automated Driving Controls. *Journal of Advanced Transportation*, 2020:1–11.

Spyropoulou, I. (2007). Simulation using gipps'car-following model—an in-depth analysis. *Transportmetrica*, 3(3):231–245.

Strickland, M., Fainekos, G., and Amor, H. B. (2018). Deep Predictive Models for Collision Risk Assessment in Autonomous Driving. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4685–4692. ISSN: 2577-087X.

Su, H., Zhong, Y. D., Dey, B., and Chakraborty, A. (2021). EMVLight: A Decentralized Reinforcement Learning Framework for Efficient Passage of Emergency Vehicles. *arXiv:2109.05429 [cs, eess]*. arXiv: 2109.05429.

Suh, Y.-H., Woo, S.-P., Kim, H., and Park, D.-H. (2019). A sim2real framework enabling decentralized agents to execute maddpg tasks. In *Proceedings of the Workshop on Distributed Infrastructures for Deep Learning*, pages 1–6.

Sukhbaatar, S., Szlam, A., and Fergus, R. (2016). Learning Multiagent Communication with Backpropagation. arXiv:1605.07736 [cs].

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning, second edition: An Introduction*. MIT Press. Google-Books-ID: uWV0DwAAQBAJ.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. arXiv:1312.6199 [cs].

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., and Vicente, R. (2015). Multiagent Cooperation and Competition with Deep Reinforcement Learning. *arXiv:1511.08779 [cs, q-bio]*. arXiv: 1511.08779.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. *CoRR*, abs/1703.06907.

Tonguz, O., Wisitpongphan, N., Bai, F., Mudalige, P., and Sadekar, V. (2007). Broadcasting in VANET. In *2007 Mobile Networking for Vehicular Environments*, pages 7–12.

Tram, T., Jansson, A., Grönberg, R., Ali, M., and Sjöberg, J. (2018). Learning Negotiating Behavior Between Cars in Intersections using Deep Q-Learning. arXiv:1810.10469 [cs, stat].

Troullinos, D., Chalkiadakis, G., Papamichail, I., and Papageorgiou, M. (2021). Collaborative Multiagent Decision Making for Lane-Free Autonomous Driving.

Tu, J., Wang, T., Wang, J., Manivasagam, S., Ren, M., and Urtasun, R. (2021). Adversarial Attacks On Multi-Agent Communication. arXiv:2101.06560 [cs].

van der horst, R. and Hogema, J. (1994). TIME-TO-COLLISION AND COLLISION AVOIDANCE SYSTEMS.

Vasirani, M. and Ossowski, S. (2012). A Market-Inspired Approach for Intersection Management in Urban Road Traffic Networks. *Journal of Artificial Intelligence Research*, 43:621–659. arXiv:1401.5851 [cs].

Vinitsky, E., Lichtlé, N., Yang, X., Amos, B., and Foerster, J. (2022). Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. *arXiv preprint arXiv:2206.09889*.

Viriyasitavat, W. and Tonguz, O. K. (2012). Priority Management of Emergency Vehicles at Intersections Using Self-Organized Traffic Control. In *2012 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–4, Quebec City, QC, Canada. IEEE.

Wakabayashi, H., Takahashi, Y., Niimi, S., and Renge, K. (2003). Traffic conflict analysis using vehicle tracking system/digital vcr and proposal of a new conflict indicator. *Infrastructure Planning Review*, 20:949–956.

Wallar, A., Zee, M. V. D., Alonso-Mora, J., and Rus, D. (2018). Vehicle Rebalancing for Mobility-on-Demand Systems with Ride-Sharing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4539–4546. ISSN: 2153-0866.

Wang, G., Niu, H., Zhu, D., Hu, J., Zhan, X., and Zhou, G. (2021a). Model: A modularized end-to-end reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:2110.11573*.

Wang, M., Wang, Z., Talbot, J., Gerdes, J. C., and Schwager, M. (2021b). Game-Theoretic

Planning for Self-Driving Cars in Multivehicle Competitive Scenarios. *IEEE Transactions on Robotics*, 37(4):1313–1325. Conference Name: IEEE Transactions on Robotics.

Wang, P., Chan, C.-Y., and de La Fortelle, A. (2018). A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1379–1384, Changshu. IEEE.

Wang, R., He, X., Yu, R., Qiu, W., An, B., and Rabinovich, Z. (2020a). Learning Efficient Multi-agent Communication: An Information Bottleneck Approach. arXiv:1911.06992 [cs].

Wang, T., Cao, J., and Hussain, A. (2021c). Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. *Transportation research part C: emerging technologies*, 125:103046.

Wang, T.-H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., and Urtasun, R. (2020b). V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, volume 12347, pages 605–621. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Wang, W., Yang, T., Liu, Y., Hao, J., Hao, X., Hu, Y., Chen, Y., Fan, C., and Gao, Y. (2019). From Few to More: Large-scale Dynamic Multiagent Curriculum Learning. *arXiv:1909.02790 [cs]*. arXiv: 1909.02790.

Wang, X. (2022). Ensuring Safety of Learning-Based Motion Planners Using Control Barrier Functions. *IEEE Robotics and Automation Letters*, 7(2):4773–4780. Conference Name: IEEE Robotics and Automation Letters.

Wang, Y., Zhong, F., Xu, J., and Wang, Y. (2022). ToM2C: Target-oriented Multi-agent Communication and Cooperation with Theory of Mind. arXiv:2111.09189 [cs].

Wang, Z., Yang, K., Li, L., Lu, Y., and Tao, Y. (2023). Traffic signal priority control based on shared experience multi-agent deep reinforcement learning. *IET Intelligent Transport Systems*, 17(7):1363–1379.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Wei, X., Tan, G., and Ding, N. (2014). Batch-Light: An adaptive intelligent intersection control policy for autonomous vehicles. In *2014 IEEE International Conference on Progress in Informatics and Computing*, pages 98–103.

Wen, L., Duan, J., Li, S. E., Xu, S., and Peng, H. (2020). Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE.

Wen, M. J., Zhao, J., and Jaillet, P. (2017). Rebalancing Shared Mobility-on-Demand Systems: a Reinforcement Learning Approach. page 6.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Wong, A., Bäck, T., Kononova, A. V., and Plaat, A. (2023). Deep multiagent reinforcement learning: challenges and directions. *Artificial Intelligence Review*, 56(6):5023–5056.

World Health Organization (2023). Road traffic injuries. Accessed: 2024-05-26.

Wu, J., Zhou, Y., Yang, H., Huang, Z., and Lv, C. (2023). Human-Guided Reinforcement Learning With Sim-to-Real Transfer for Autonomous Navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14745–14759. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Xiang, H., Xu, R., Xia, X., Zheng, Z., Zhou, B., and Ma, J. (2023). V2XP-ASG: Generating Adversarial Scenes for Vehicle-to-Everything Perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3584–3591.

Xie, J., Liu, Y., and Chen, N. (2023). Two-Sided Deep Reinforcement Learning for Dynamic Mobility-on-Demand Management with Mixed Autonomy. *Transportation Science*, 57(4):1019–1046. Publisher: INFORMS.

Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., and Ma, J. (2021). OpenCDA:An Open Cooperative Driving Automation Framework Integrated with Co-Simulation. arXiv:2107.06260 [cs].

Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.-H., and Ma, J. (2022a). V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer. arXiv:2203.10638 [cs].

Xu, R., Xiang, H., Xia, X., Han, X., Li, J., and Ma, J. (2022b). OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication. arXiv:2109.07644 [cs].

Xue, W., Qiu, W., An, B., Rabinovich, Z., Obraztsova, S., and Yeo, C. K. (2022). Mis-spoke or mis-lead: Achieving Robustness in Multi-Agent Communicative Reinforcement Learning. arXiv:2108.03803 [cs].

Yadav, P., Mishra, A., and Kim, S. (2023). A Comprehensive Survey on Multi-Agent Reinforcement Learning for Connected and Automated Vehicles. *Sensors*, 23(10):4710. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. (2018). Mean Field Multi-Agent Reinforcement Learning. *arXiv:1802.05438 [cs]*. arXiv: 1802.05438.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. (2021). The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv:2103.01955 [cs]*.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C.-J. (2021a). Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. arXiv:2003.08938 [cs, stat].

Zhang, K., De La Fortelle, A., Zhang, D., and Wu, X. (2013). Analysis and Modeled Design of One State-Driven Autonomous Passing-Through Algorithm for Driverless Vehicles at Intersections. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 751–757.

Zhang, K., Sun, T., Tao, Y., Genc, S., Mallya, S., and Basar, T. (2020). Robust multi-agent reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 33:10571–10583.

Zhang, L., Zhang, R., Wu, T., Weng, R., Han, M., and Zhao, Y. (2021b). Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5435–5444.

Zhang, R. and Pavone, M. (2014). Control of Robotic Mobility-On-Demand Systems: a Queueing-Theoretical Perspective. *arXiv:1404.4391 [cs]*. arXiv: 1404.4391.

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)*, pages 6227–6232, Boston, MA, USA. IEEE.

Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE.

Zheng, L., Ismail, K., and Meng, X. (2014). Shifted Gamma-Generalized Pareto Distribution model to map the safety continuum and estimate crashes. *Safety Science*, 64:155–162.

Zhou, M., Luo, J., Villella, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., Huang, A. C., Wen, Y., Hassanzadeh, K., Graves, D., Chen, D., Zhu, Z., Nguyen, N., Elsayed, M., Shao, K., Ahilan, S., Zhang, B., Wu, J., Fu, Z., Rezaee, K., Yadmellat, P., Rohani, M., Nieves, N. P., Ni, Y., Banijamali, S., Rivers, A. C., Tian, Z., Palenicek, D., Ammar, H. b., Zhang, H., Liu, W., Hao, J., and Wang, J. (2020a). SMARTS: Scalable Multi-Agent Reinforcement Learning Training School for Autonomous Driving. *arXiv:2010.09776 [cs, eess]*.

Zhou, M., Luo, J., Villella, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., Huang, A. C., Wen, Y., Hassanzadeh, K., Graves, D., Chen, D., Zhu, Z., Nguyen, N., Elsayed, M., Shao, K., Ahilan, S., Zhang, B., Wu, J., Fu, Z., Rezaee, K., Yadmellat, P., Rohani, M., Nieves, N. P., Ni, Y., Banijamali, S., Rivers, A. C., Tian, Z., Palenicek,

D., bou Ammar, H., Zhang, H., Liu, W., Hao, J., and Wang, J. (2020b). Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving.

Zhou, W., Chen, D., Yan, J., Li, Z., Yin, H., and Ge, W. (2022). Multi-agent Reinforcement Learning for Cooperative Lane Changing of Connected and Autonomous Vehicles in Mixed Traffic. *Autonomous Intelligent Systems*, 2(1):5. arXiv: 2111.06318.

Zhu, C., Dastani, M., and Wang, S. (2022). A Survey of Multi-Agent Reinforcement Learning with Communication. arXiv:2203.08975 [cs].

Zhu, F. and Ukkusuri, S. V. (2015). A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:363–378.

Zhu, H., Yu, J., Gupta, A., Shah, D., Hartikainen, K., Singh, A., Kumar, V., and Levine, S. (2020a). The ingredients of real-world robotic reinforcement learning. *arXiv preprint arXiv:2004.12570*.

Zhu, M., Wang, Y., Pu, Z., Hu, J., Wang, X., and Ke, R. (2020b). Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transportation Research Part C: Emerging Technologies*, 117:102662.

Zohdy, I. H., Kamalanathsharma, R. K., and Rakha, H. (2012). Intersection management for autonomous vehicles using iCACC. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1109–1114. ISSN: 2153-0017.