Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Safe Lane-changing in Connected Autonomous Vehicles using Safety Supervisors

Lalu Prasad Lenka

August 2022

Supervisor: Prof. Mélanie Bouroche

A dissertation submitted to University of Dublin, Trinity College in partial fulfilment of the requirements for the degree of Master of Science in Computer Science (Data Science)

Declaration

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

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

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

Signed: Lalu Prasad Lenka Date: 19/08/2022

Abstract

Connected autonomous driving has piqued the curiosity of the research community over recent years due to its potential to provide driving assistance and reduce traffic congestion, among other benefits. Despite promising advancements, safe lane-changing remains a significant challenge for connected autonomous vehicles (CAVs), particularly in mixed and dynamic traffic conditions. Investigation of the state-of-the-art papers on motion planning for CAVs suggests a gap in research for safety supervisor techniques for lane changing algorithms in CAVs.

This paper uses multi-agent reinforcement learning (MARL) to model lane-changing in connected autonomous vehicles in mixed traffic scenarios, i.e., with human-driven vehicles (HDVs) on the road. Parameter sharing and replay buffer are employed to motivate cooperative behaviour and collaboration among CAVs. An OpenAI gym-like environment highway-env is developed and modified to simulate the lane changing in CAVs. In addition, various state-of-the-art safety supervisor techniques for reinforcement learning (RL) approaches are analysed, and their applicability to designing safer MARL for lane changing of connected autonomous vehicles (MARL-CAV) is examined. Comprehensive analysis and experimental results show that integrating some promising safety supervisors to MARL for lane changing in CAVs is challenging, and none of the existing safety supervisor techniques can be directly applied to MARL-CAV as these safety techniques require prior knowledge of unsafe states and recovery policies.

Acknowledgements

I express my sincere gratitude to my supervisor, Assistant Prof. Melanie Bouroche. I am grateful for her patience, suggestions and continued guidance throughout this dissertation's research, implementation and writing phase. This project could not have been completed without her supervision. I want to convey my deep gratitude to Prof. Melanie for her unending support that enabled me to drive this dissertation.

Also, I would to thank my mummy, papa and brothers for their continuous support and unfaltering faith in me. This master's course and dissertation would not have been possible without them.

Lalu Prasad Lenka
MSc. Computer Science (Data Science)
Trinity College Dublin

Contents

List of Figures

List of Tables

1 Introduction

Over the past few years, the buzz surrounding self-driving automobiles has grown significantly. The automotive industry is undergoing significant technical transformation at the dawn of the twenty-first century, as several significant automotive and technology organisations attempt to create and advance autonomous vehicle technologies.

Google launched their autonomous driving project "Waymo" in 2009 (Google 2022) and has recently partnered with Jaguar to develop self-driving vehicles. Tesla's Autopilot is arguably the most advanced self-driving technology today (*Tesla's Autopilot* 2022). Many leading automobile organisations like Mercedes Benz, General Motors etc., are actively researching, developing and testing autonomous vehicle technologies (Innovation 2022, Motors 2022).

Improvements in tra□c mobility, eco-friendly vehicles, fewer fossil fuels and safer roads are among the benefits promised by the autonomous vehicle industry. Therefore autonomous vehicle technology has spurred a significant drive in the research of autonomous driving technology in academia and industry.

> *"The key with autonomous is the whole ecosystem. One of the keys to having a truly fully autonomous is vehicles talking to each other." - Mary Barra, GM CEO*

A connected autonomous vehicle (CAV) is an autonomous vehicle linked to other vehicles or roadside units (Paret et al. 2022). Digital connectivity is predicted to positively impact driving comfort, tra□c e□ciency, and safety. CAV technology is expected to generate jobs and create a new market for connected, automated, and cooperative driving.

Therefore, connected autonomous vehicle (CAV) technology is a major focus of governments worldwide. In recent times substantial progress has been made in developing CAV technologies. In 2016 the European Commission adopted a European Strategy on Cooperative Intelligent Transport Systems (C-ITS), a milestone toward cooperative, connected and automated mobility (Commission 2022). With a focus on adopting and deploying CAV infrastructure on a large scale, "Realising connected vehicle implementation" and "advancing automation" are the two primary strategic priorities in USDOT's ITS 2015-2019 strategic plan (USDOT 2022).

Autonomous car driving is one of the most important topics in the contemporary artificial intelligence (AI) research field with a potentially huge impact (Darapaneni et al. 2021). Over the last decade, there has been a lot of research on using AI algorithms to develop autonomous vehicle software (Haydari & Yılmaz 2022*a*, Takehara & Gonsalves 2021, Darapaneni et al. 2021, Liu 2020, Okuyama et al. 2018, Kulkarni et al. 2018).

> *"Self-driving cars are the natural extension of active safety and obviously something we should do" – Elon Musk.*

Safety in intelligent transportation systems is the first strategic goal of the US department of transportation (DOT 2022). Though there has been great advancement in AV technology over the past decade that has made this possible, the number of tra□c accidents involving autonomous vehicles has increased in recent years (Dixit et al. 2016, Favarò et al. 2017).

A major problem with automated driving at its current stage of development is that it is not yet reliable and safe (Martens & van den Beukel 2013). At the time of writing this dissertation, all the autonomous vehicles on the market are semi-automatic. Therefore, when automated driving fails or is limited, i.e., the onboard computer algorithms cannot make a safe decision, it expects the human driver to take over, and if there is a delay in this transition, it leads to an accident.

This motivates the development of safer software that uses artificial intelligence to drive the vehicle safely.

This chapter introduces autonomous vehicles and connected autonomous vehicles in more detail and provides a background of the science behind their functioning and decision-making. This is followed by motivation for this work, the research question it addresses and its contribution. The chapter ends with a brief outline of the thesis layout.

1.1 Background

This section briefly discusses the background about autonomous and connected autonomous vehicles (1.1.1). Then it discusses how motion planning works for CAVs (1.1.2) and at the end briefly talks about safe motion planning in CAVs (1.1.3).

1.1.1 Introduction to Connected Autonomous Vehicles

Autonomous Vehicles (AV)

An autonomous vehicle can be defined as a vehicle capable of travelling unassisted anywhere and at any time, with no restrictions and without the help or even the presence of a driver(Paret et al. 2022).

Driving requires a variety of functions beyond decision-making, most notably involving information acquisition in support of environment perception. If all of those other functions were also performed by self-contained systems in the vehicle, its automation system could legitimately be considered to be “autonomous”. If the vehicles use communications with the infrastructure or other vehicles to acquire information or to negotiate manoeuvres, they have “cooperative” rather than “autonomous” automation systems (Shladover 2018).

Autonomous vehicles can be classified based on the level of automation capability. SAE International (formerly the Society of Automotive Engineers) has developed a five-level classification of automated driving systems, which is useful for distinguishing the capabilities available at each level. SAE (2022) classifies autonomous driving at five levels based on driver support features, automated driving features and human responsibilities in the driver’s seat.

Connected Vehicles (CV)

Connected vehicles possess the ability to knit vehicles and infrastructure elements into a well-integrated transportation system and hence, support a wide range of Intelligent Transportation Systems (ITS) applications. (Shladover 2018)

The connectivity in connected vehicles can be of several types, namely V2V (vehicle to vehicle), V2I (vehicle to infrastructure), I2V (Infrastructure to vehicle), V2P (vehicle to pedestrian), and V2X (Vehicle to anything). This section focuses on some aspects of V2V (vehicle to vehicle) communication.

Shladover (2018) stated that V2V connectivity can enable applications such as :

- Cooperative collision warnings and hazard alerts, as tested in the Safety Pilot Model Deployment
- Cooperative collision mitigation or avoidance, incorporating active braking
- Cooperative adaptive cruise control, with a tighter vehicle following control than conventional adaptive cruise control and enhanced tra□c flow stability
- Close-formation automated platooning, enabling aerodynamic drafting and lane capacity increases
- Automated manoeuvre negotiation at merging locations or intersections

These time-critical and safety-critical applications need very low latency and high-reliability communications. For most of these applications, the communicated data is used to augment the data acquired by onboard remote sensors, which remain the primary source of data about time-critical and safety-critical conditions.

Connected Autonomous Vehicles (CAVs)

The CV and AV developments have been proceeding along somewhat independent paths, but there should be a strong synergy between them. They provide complementary contributions to improving transportation system performance and safety. Connectivity integrates the vehicles and the infrastructure into a system whose performance can be adjusted to satisfy various societal goals (Shladover 2018).

An AV without connectivity is like a sensory-deprived driver who can only see the surrounding environment but cannot communicate with other drivers by any currently available means and therefore lacks full knowledge of that environment.

The CV communication links provide information that AV sensors cannot see (maneuver commands issued to other vehicles, faults on other vehicles, accelerations of those vehicles) and can also provide information about vehicles beyond the immediate line of sight. This is crucial for dampening shock waves in tra□c, which means that it has a strong impact on roadway capacity, energy consumption, emissions and safety.

Autonomous vehicles (AVs without connectivity) can only see the vehicles immediately adjacent to them, which means that they cannot see the vehicles several positions further ahead and therefore cannot anticipate acceleration or braking transients in tra□c. In this regard, they are worse than human drivers and produce less stable car following than human drivers (Milanes 2014). When V2V communication is added to provide information about the motions of the vehicles further ahead, the disturbances are dampened, and tra□c flow becomes much smoother.

V2V connectivity also has important safety implications. At the most elementary level, many secondary rear-end crashes on highways could be eliminated if V2V communications could alert drivers approaching a congestion queue that the tra□c ahead is going much slower or stopped. If any vehicle encounters a hazard or su□ers an internal fault, it can communicate that information to its neighbours to start their safety recovery responses immediately, even before their own sensors detect any problems. If a vehicle su□ers such a severe fault that its communication system is disabled, the sudden loss of communications from that vehicle also informs its neighbours about a problem, providing a fail-safe backup.

This paper examines the current state-of-the-art approaches for lane-changing in Connected Autonomous Vehicles (CAVs).

1.1.2 Motion Planning for Connected Autonomous Vehicles

The motion planning problem is the task of navigating an autonomous vehicle to its destination safely and comfortably while following the rules of the road.

At a high level, this task can be decomposed into a hierarchy of optimisation problems. Each of which will have di erent constraints and objectives.

The first part of motion planning in autonomous vehicles includes estimating the vehicle's state, localising its surrounding vehicles, obstacles, lane detection etc. The sensors that help with this are an accelerometer, Gyroscope, LIDAR (Light Detection and Ranging) and GNSS (Global Navigation Satellite Systems). Sensor Fusion using Kalman Filter is used to fuse the data from several sensors to estimate the state of any robotic system in real-time.

The second part is modelling the vehicle's motion. Di erent mathematical models are used as suitable control-oriented models to represent vehicles.

Kinematic Bicycle Modeling is used to capture vehicle motion in normal driving conditions. This model is used to design controllers for autonomous vehicles. It captures vehicle motion with steering rates and velocity inputs. Dynamic Modeling is used to get higher fidelity predictions than are possible with kinematic models. This higher fidelity, however, comes at the cost of higher computational complexity. So both kinematic and dynamic models have their uses in self-driving development.

The third and fourth parts of motion planning are behaviour and path planning. These are the most important parts of motion planning, they aim to decide when it is safe to proceed considering the static and dynamic constraints. Behaviour planning focuses on high-level decision-making required to follow the rules of the road and recognise which manoeuvres are safe to make in a given driving scenario. The path planner finds a kinematically feasible and collision-free path.

The motion planning for connected autonomous vehicles (CAVs) di ers from motion planning for autonomous vehicles (AVs) because the CAVs can connect with nearby vehicles and roadside units using communication technology like dedicated short-range communication (DSRC). Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) can be established using DSRC, which helps in cooperative-adaptive cruise control as multiple connected vehicles can communicate and take cooperative action (Rana & Hossain 2021). Vehicles can also share information with each other enabling collective learning enabling decentralized planning with shared information.

1.1.3 Safety in Motion Planning for Connected Autonomous Vehicles

The safety in connected autonomous vehicle (CAvs) refers to the ability of the CAV system to function normally and to ensure safety of passengers and other road users. This includes the following criteria:

- CAV should ensure smooth manoeuvres in road like lane following, lane changing, merging etc.
- CAV should avoid collision with nearby vehicles, pedestrian or other obstacles in road.
- CAV should not change the speed very aggressively which might be uncomfortable for the passengers.
- CAV should follow the tra□c rules.

Absence of any one or more criteria can be considered unsafe motion planning. Perhaps the biggest obstacle to Artificial Intelligence (AI) becoming a more popular replacement for traditional engineering solutions in safety-critical applications is safety. One of the underlying problems is the fact that formal guarantees are currently lacking in neural networks, making them generally unreliable. In AI approaches like Reinforcement Learning there is no formal guarantee that the agent will learn a given task safely. This is because the agent visits unsafe states during exploration phase of reinforcement learning algorithm. However, in real-world safety-critical systems would require an exploratory algorithm to ensure safety.

This dissertation aims to review and analyse external safety techniques that can be applied to AI algorithms for connected autonomous vehicle manoeuvres and can ensure with high probability that the vehicle stays safe.

1.2 Motivation

Connected and automated vehicle (CAV) technologies have been developed throughout the years to improve traffic safety, mobility, and environmental impacts. The lane-changing feature is one of the CAVs technologies that make it possible.

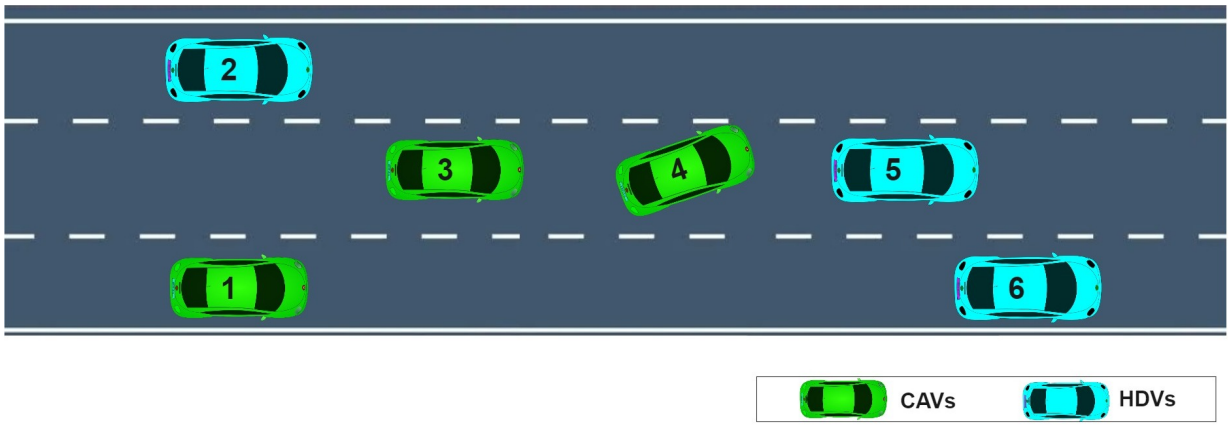


Figure 1.1: Illustration of the considered lane changing in mixed traffic scenario. CAVs (green) and HDVs (blue).

CAVs must communicate with other road users, who can be cooperative or aggressive, alert or inattentive. Such disparities in traits can result in a wide range of interacting behaviours. As a result, CAVs must be aware of such uncertainties when planning their behaviours to achieve safe and efficient autonomous driving. The CAVs need to consider the action of other human-driven vehicles and autonomous vehicles during lane changing, making this a difficult task to automate.

Several variants of AI algorithms based on algorithms like deep reinforcement learning, game theory and rule-based, have been proposed for efficient lane changing among CAVs. However, most papers do not discuss the safety constraints in these safety-critical lane-changing manoeuvres. I wish to compare the current state-of-the-art algorithms based on their implementation's safety challenges and limitations to avoid crashing with static and dynamic obstacles. This would help me understand the trade-off between the safety (of driver and vehicle) and efficiency (in lane-changing) aspects of these algorithms. Furthermore, I also wish to investigate novel variant of safety approaches for the AI algorithms which prioritises safety in lane-changing manoeuvres.

Safe Reinforcement learning aims to create a safe learning algorithm while testing and during training. Most of the implementation of reinforcement learning introduce safety either through reward function or through an external supervisor which guides the agent during the training process. The literature on the later approach is limited, and I wish investigate the possibility of using external safety supervisors to simulate safe lane changing in CAVs.

Two questions which motivate this dissertation are:

- How do make safer multi-agent reinforcement algorithms for lane changing in CAVs?
- How to ensure safety with scalability, i.e. as the penetration rate of CAVs increases, how can we ensure safety?

1.3 Research Question

The purpose of this project is to answer the following research question:

“Can external safety supervisors help enhance the safety of multi-agent RL algorithm for lane changing in connected autonomous vehicles?”

From this question, the following research objectives were defined:

- Understand state-of-art AI algorithms proposed for e□cient lane changing.
- Compare and evaluate di□erent AI algorithms for lane changing and how they ensure the safety of both drivers and connected autonomous vehicles.
- Analyse various external safety supervision techniques to understand their functioning, assumptions and limitations.
- Compare and evaluate these safety supervision techniques for lane changing in connected autonomous vehicles.

1.4 Contribution

The main contributions and technical advancements of this dissertation are as follows:

1. Formulated mixed-tra□c lane changing problem (with CAVs and HDVs coexisting in di□erent lanes) as cooperative decentralised multi-agent reinforcement learning (MARL) problem with custom reward function specific for lane-changing task.
2. Investigated multiple state-of-the-art safety techniques that enable safe exploration in the multi-agent setting.
3. Analysed the functioning, assumptions and limitations of these safety techniques to verify their suitability in MARL for lane changing in CAVs.

1.5 Overview of Thesis

The structure of the thesis is explained in this section. Chapter 1 starts with a brief background of connected autonomous vehicles and motion planning in connected autonomous vehicles. The motivation, research question and contribution are presented after the background. Next, Chapter 2 discusses the first recent research related to motion planning and lane changing in CAVs. After that, previous works on safety approaches for lane-changing in CAVs are discussed. Then the focus is on recent research on reinforcement learning and safety approaches in reinforcement learning. Next, Chapter 3 presents the formulation of lane changing in CAVs as a multi-agent RL (MARL) problem. Chapter 4 provides a thorough analysis of external safety techniques for MARL. Chapter 5 covers the implementation details of the simulation platform used to simulate MARL for lane changing in CAVs. In Chapter 6, an evaluation of the MARL-CAV model is performed, and the results are discussed in detail. Finally, the conclusion of the dissertation was provided in Chapter 7, and it also covers the challenges, future work and the author's reflection on this project work.

2 Literature Review

This chapter discusses the various papers I have studied in filed of artificial intelligence algorithms and autonomous vehicles. This chapter has helped drive the discussions in this dissertation to answer the stated the research questions. It first discusses the state of the art in motion planning for CAVs (Section 2.1), then focuses on lane changing approaches (Section 2.2), before discussing the existing approaches to safety for CAV lane changes (Section 2.3). Then it discusses the recent advancements in multi-agent reinforcement learning (MARL) (2.4). Finally, this chapter discusses the external safety approaches for MARL applied in recent research.

2.1 Motion Planning for CAV

Motion planning generates the driving manoeuvre to be performed by the vehicle. Various manoeuvres in roads like maintaining stable speed, deceleration, acceleration, lane following, lane merging, lane changing and obstacle scenarios like static and dynamic obstacles a□ect the driving behaviour and are modelled into the vehicle behaviour planning. Over the years, several approaches have been studied to develop motion planners for autonomous vehicles.

In the literature, there are four prevailing architectures to solve the behavioural planning problem: Finite State Machines, rule-based, game theory and machine learning. These are discussed in turn.

2.1.1 Finite State Machines

Finite State Machines are a computational model for systems whose output depends on the entire history of their inputs. They are composed of states representing the vehicle's required manoeuvres and transitions dictating how the state should evolve depending on the inputs. States are based on the perception of their surroundings. Bae et al. (2020) used a motion finite state machine to plan actions like a left turn, right turn, and lane following and control finite state machine to plan actions like stop, acceleration, and deceleration. Chen et al. (2019) proposed a motion planner for CAVs where the motion planner creates a finite-state machine (FSM) to decide the best driving strategies while considering oncoming vehicles. Temporal

windows are constructed using information pieces from the connected vehicle, particularly vehicle location and speed, to control the driving states' transition of the FSM.

2.1.2 Rule Based

- These types of systems consist of a hierarchy of rules, where the hierarchy represents the relative importance of each rule. Each rule may correspond to a rule of the road, such as stopping at a red light, or a driving best practice such as maintaining a two-second gap between the ego-vehicle and a leading vehicle.

2.1.3 Game Theory

- The basic premise of game theory is that players maximize their individual payo□s to outperform opponents' strategies and produce better results. The autonomous driving vehicle is an agent in a game, and so are the nearby cars. Each vehicle has a certain goal, such as to go at a certain pace or stay in a particular lane. The agents must also adhere to restrictions, such as following speed limits and avoiding collisions with other vehicles.
- Fisac et al. (2019) presented a game-theoretic framework for hierarchical trajectory planning for autonomous vehicles interacting with human-driven vehicles on the road.
- Le Cleac'h et al. (2021) proposed a game theory-based approach of using constraint multi-player dynamic games to handle trajectory-optimization problems for multiple autonomous cars in a freeway merging scenario. Ji & Levinson (2020) reviewed multiple papers that used game theory approaches for lane changing in autonomous vehicles.

2.1.4 Machine Learning - Deep Learning (DL) and Reinforcement Learning (RL)

- Motion planning for autonomous vehicles is a complex task due to the nonlinear and stochastic of complex transportation systems. A vehicle manoeuvring task can involve consideration of many dynamic variables like speed (and acceleration) of neighbouring vehicles, headway, i.e. distance from the front vehicle, obstacles in the road (pedestrian, road under construction), and tra□c sign detection, and road structure etc. Therefore, it is a complicated task with a lot of data involved. Machine learning algorithms tend to perform better in tasks involving a lot of data.
- Generally, rule-based algorithms are designed based on the author's knowledge, but they might not scale in real-life scenarios. Therefore, rule-based or finite state machines might behave wrongly in a completely new scenario; hence machine learning algorithms like

deep learning and reinforcement learning are more prominently used in motion planning and other aspects of intelligent transportation systems.

- Deep learning (DL) involves training deep neural network models using a lot of data to map certain inputs (like surrounding vehicles data, traffic sign image data etc.) to outputs (like steering angle, acceleration).
- Chen et al. (2020) used deep recurrent neural networks to predict the velocity of nearby vehicles and generate driving trajectories for ego-vehicle using polynomial curves.
- Li et al. (2019) used Long Short Term Memory, a recurrent neural network, to learn traffic patterns and predict short-term traffic status.
- Veres & Moussa (2020) reviewed several papers where DL has been successfully used in various tasks like traffic sign control, navigation, traffic flow prediction and route prediction. However, Deep Learning has been employed for various applications in intelligent transportation systems (ITS). Another type of machine learning, i.e. Reinforcement Learning, is more widely used for motion planning.
- Reinforcement learning is the process of determining an optimal decision-making policy that maximizes some reward for the agent. The reinforcement learning process requires the agent to perform actions in an environment often given by simulation. This agent is then rewarded according to its interaction with the environment, allowing it to converge to an optimal policy through successive interactions.
- Haydari & Yılmaz (2022*b*) reviewed several papers that have used different reinforcement learning techniques to plan the motion of autonomous vehicles. The RL-based ITS controllers dominate the actor-critic and Q-learning-based Deep Q Networks (DQN) and Deep Deterministic Policy Gradient (DDPG) algorithms. Deep RL approaches are favoured over normal RL methods for high-dimensional state spaces. Policy-based deep RL approaches are better suited for continuous action spaces than value-based deep RL methods in terms of action space (Haydari & Yılmaz 2022*b*).

2.1.5 Summary

From the discussion in this section, it can be inferred that machine learning approaches like reinforcement learning are more widely studied and more suitable for motion planning in autonomous and connected autonomous vehicles.

2.2 Lane changing approaches for CAV

This section focuses on a more specific use case of motion planning in autonomous and connected autonomous vehicles. Lane changing remains a challenging task where the autonomous

vehicle has to predict the actions of neighbouring vehicles while changing lanes to avoid a collision. Furthermore, an autonomous vehicle is supposed to undertake the lane change manoeuvring without aggressive acceleration or deceleration and smoothly to ensure the comfort of the passengers.

Various approaches for lane changing in connected autonomous vehicles (CAVs) are discussed and compared in this section.

2.2.1 Rule-based approaches

Some papers use rule-based approaches. Li et al. (2018) presented lane changing task among CAVs as an objective function with various constraints (to avoid collision). A step-by-step computational framework is proposed, which divides the lane-changing task into sub-problems, each with a collision-avoidance constraint. Orthogonal collocation direct transcription with interior-point method is used to sequentially solve each sub-task until an optimal solution which satisfies all constraints is reached. Zheng et al. (2020) used modified version of Minimizing Overall Braking Induced by Lane Changes Model (MOBIL) (Kesting et al. 2010), a rule-based lane changing algorithm. They proposed a cooperative lane-changing strategy to improve tra□c operation and safety at a diverging area nearby a highway o□-ramp in an environment with connected and automated vehicles (CAVs). The road towards the o□-ramp is divided into "Discretionary lane change zone" and "Mandatory/Cooperative lane change zone". The rules are defined such that the safe lane changing gaps are proactively created to make vehicles move into the target lane by cooperative vehicle deceleration on the target lane.

2.2.2 Game theoretic approaches

Game theoretic (GT) approaches are common for planning lane-changing in connected autonomous vehicles. Lin et al. (2019) proposed a cooperative lane changing strategy using a transferable utility games framework which allows vehicles to engage in transactions where gaps in tra□c are created in exchange for monetary compensation. The proposed approach is best suited to discretionary lane change manoeuvres. Based on CAV vehicle-to-vehicle (V2V) communication functionality, the paper presents a lane-changing mechanism that allows vehicles to purchase right of way or compensate other vehicles for allowing them to change lanes. Liao et al. (2021) proposed a game theory-based ramp merging strategy with a decentralized algorithm, providing the optimal merging sequence and associated speed trajectory for each CAV in the *mixed tra c*. In the simulation performed in the paper, the car-following and lane changing are performed by default models in SUMO and Unity platforms. The proposed algorithm for ramp merging predicts a conflict scenario between an ego CAV with CAV/legacy vehicles and uses game theory to determine the merge sequence and acceleration controls for the ego CAV. The overall cost function in the algorithm is a function of collision risks, comfort

Table 2.1: Different game theory approaches with their features

Model	Features
The empirical GT model	is reliable due to being calibrated from real scenarios
Classic Nash equilibrium model	achieves the best personal choices with complete information
The incomplete information game model	separates LC into two types: mandatory and discretionary
The sequential game model	supposes one reacts first and the other responds later
EGT-based model	promotes the cooperative rate in groups progressively

(of passengers and drivers) and mobility (of traffic).

Ji & Levinson (2020) reviewed the development of game-theoretic models for lane changing over three decades; these algorithms can be classified according to their different methodologies and objective function. Firstly, the paper has reviewed the utility (which the drivers may gain from their actions) function and how it was improved over time. The most recent papers generally frame utility as a function of safety, time headway, space, and travel time. Table 2.1 shows varying game theory algorithms used in lane changing:

Most of the papers that used the game theory approach for lane changing designed the cost/utility function of lane change manoeuvres using these seven components, namely, safety, equilibrium, control, travel efficiency, route choice, lane preference, and willingness (to change lanes), to design cooperative controllers.

In summary, the basic form GT-based models have developed from simple static forms with complete information that consider few factors to complicated dynamic forms with incomplete information that cover multiple factors. They all demonstrate the feasibility of GT in revealing human interaction and decision-making processes. Evolutionary game theory (EGT) based lane changing algorithms can be used to build more cooperative and selfless lane-changing algorithms.

2.2.3 Reinforcement learning approaches

Fu et al. (2020) modelled the lane-changing problem as a deep reinforcement learning process to learn the optimal lane-changing strategy through a deep deterministic policy gradient (DDPG) algorithm. They also proposed a collective learning framework to use the collective intelligence of CAVs to improve the performance of autonomous lane-changing strategies.

Through knowledge transfer, on the one hand, the privileged information reduces the unnecessary action space, thereby accelerating the learning process. On the other hand, through the continuous accumulation of knowledge and the improvement of the DRL model, the CAVs in the whole network can learn the optimal autonomous driving model.

Zhou et al. (2021) proposed a decentralized cooperative multi-agent reinforcement learning algorithm with an actor-critic policy for lane changing among CAVs. The paper considers a lane-changing decision-making problem statement having multiple AVs in a mixed-traffic highway environment with varying levels of traffic densities. The proposed algorithm features a novel local reward design incorporating safety, efficiency and passenger comfort and a parameter-sharing scheme to foster inter-agent collaborations.

The proposed algorithm was thoughtfully designed to overcome the limitations of other state-of-the-art RL algorithms for lane changing. It achieved stable performance for varying degrees of traffic densities and varying levels of aggressiveness in HDVs. It also showed relatively higher stability and success when compared with other state-of-the-art RL algorithms like *Multi-agent Proximal Policy Optimization (MAPPO), Multi-agent actor-critic using Kronecker-Factored Trust Region (MAACKTR), Multi-agent Deep Q-Network (MADQN).*

Ye et al. (2020) proposed an automated mandatory lane change strategy using proximal policy optimization-based deep reinforcement learning, showing great advantages in learning efficiency while maintaining stable performance. Unlike previous studies which used deep Q learning, they used the safe proximal policy optimization (PPO)-based deep reinforcement learning method, which combines the policy with a safety intervention module. The simulation is done on SUMO with the relatively simple scenario as the state space of the RL environment is composed of a total of 21 continuous state variables from both the ego-vehicle and its surrounding five vehicles. IDM and Mobil are used for HDVs.The reward function consists of comfort (lateral jerks and longitudinal jerks), efficiency (travel time and relative distance to the target lane;) and safety (risk of collisions and near collisions). The Safety intervention module identifies “catastrophic” actions and returns a negative penalty reward.

Chen et al. (2022) proposed an efficient and scalable MARL framework that can be used in dynamic traffic where the communication topology could be time-varying. Parameter sharing and local rewards are exploited to foster inter-agent cooperation while achieving great scalability. An action masking scheme improves learning efficiency by filtering out invalid/unsafe actions at each step. In addition, a novel priority-based safety supervisor is developed to significantly reduce collision rates and greatly expedite the training process.

2.2.4 Summary

From the papers studied in Sections 2.1 and 2.2 it can be inferred that reinforcement learning is most widely used to develop approaches for motion planning in connected autonomous

vehicles.

2.3 Safety approaches in Lane changing for CAV

As of now, the lane-changing approaches have been discussed. The rule-based, game theoretic and reinforcement learning (RL) approaches are common, with RL approaches being more extensively used. This section discusses the safety aspect of these approaches in detail.

2.3.1 Rule-based

Khayatian et al. (2021) have discussed that most lane-changing algorithms have a safety bu□er around each vehicle to cover for localization and trajectory tracking uncertainties. The paper proposes a trajectory-based definition for RSS rules that works in all situations, including merges, intersections, and unstructured roads. The proposed algorithm can be integrated with any motion planner algorithm. The paper presents a cooperative driving algorithm for CAVs based on proposed RSS rules. Zheng et al. (2020) proposed cooperative strategies to enable safe lane changing and used a set of metrics, namely The total travel time (TTT), Time Exposed Time-to-collision (TET) and the modified Time Integrated Time-to-collision (TIT), as ways to evaluate the safety of lane changing algorithms. A lower TET value indicates a safer situation. TIT calculates the entity of the TTC lower than the threshold, expressing the severity associated with safety-critical situations.

2.3.2 Game Theory Based

All the game-theoretic approaches discussed in the previous section reflcapproaches referred to (Lin et al. 2019, Liao et al. 2021, Ji & Levinson 2020) the safety of agents and were taken into account by customizing the utility/cost function. There were no other approaches discussed to inculcate safety into their approaches. This shows a gap in research where we can introduce external knowledge for adding safety into game-theory approaches.

2.3.3 Reinforcement Learning Based

Fu et al. (2020) discussed blockchain-based collective learning framework for lane-changing in CAVs and introduced headway distance (i.e. the distance between ego-vehicle and leading vehicle) into reward function to make the lane changes safer.

Ye et al. (2020) used proximal policy optimization (PPO) for lane-changing in autonomous vehicles and introduced collision reward to penalize unsafe actions, and used a safety intervention module to label the output action from the algorithm as “catastrophic” or “safe”. However, the details of the safety intervention module were not discussed.

Zhou et al. (2021) proposed multi-agent actor-critic RL for cooperative lane changing among autonomous vehicles used headway distance and collision penalty in reward function to add safety into their design.

Chen et al. (2022) also used headway distance and collision penalty in reward function to add safety but also proposed a novel priority-based safety supervisor which predicts the action of neighbouring vehicles to enable safer decisions. This was the only paper which used an external safety technique.

Surprisingly, most papers introduce the safety aspect through utility or reward functions in game-theoretic and RL approaches. We can infer a research gap as external safety techniques for lane-changing algorithms for connected autonomous vehicles are less prevalent. This motivates me to work towards using external safety techniques in Game theory and Reinforcement Learning to develop safer RL-based lane-changing algorithms for connected autonomous vehicles.

2.3.4 Summary

In section 2.2 the conclusion was that reinforcement learning techniques are most widely used for lane changing in CAVs. In this section it was observed that in recent literature, the safety in MARL was achieved strictly through reward functions. In the following sections, the focus will be on multi-agent reinforcement learning techniques and exploring techniques that can be used to develop external safety supervisors for RL algorithms.

2.4 Multi-agent Reinforcement Learning (MARL)

2.4.1 Overview

This section focuses on reinforcement learning (RL) and how multi-agent reinforcement (MARL) works. The section also discusses challenges in MARL and which category of RL algorithms works better for developing MARL.

Most successful RL applications, e.g., the games of Go and Poker, robotics, and autonomous driving, involve the participation of more than one single agent, which naturally falls into the realm of multi-agent RL (MARL) (Zhang, Yang & Başar 2019).

Generally, reinforcement learning algorithms can be divided into model-based and model-free RL. Model-based means the RL agent has access to or learns a model of the environment, whereas, in model-free RL, the agent doesn't know about the state transition function of the environment. In simple words, it doesn't know for certain what the state of the environment will be when an action is taken. The key benefit of having a model-based implementation is that it enables the agent to plan by anticipating future events, analyzing the outcomes of

various options, and making explicit decisions regarding its options. The main drawback is that the agent frequently does not have access to a ground-truth model of the environment (Achiam 2018). For a lane-changing scenario of CAVs, there is no uncertainty about how the environment will change by the action of one or more agents. Therefore, model-free RL approaches are studied and applied in this dissertation.

Furthermore, the two main approaches to represent and train agents in model-free RL are Q-learning and policy optimization. Deep Q learning algorithms try to learn the optimal Q function $Q(s, a)$ with a function approximator $Q_\theta(s, a)$. It generally uses an objective function based on Bellman equation. To the contrary, Policy optimization algorithms try to directly learn the policy $\pi_\theta(a|s)$ by optimizing parameters theta by using gradient ascent on performance objective $J(\pi_\theta)$ (Achiam 2018).

The primary advantage of policy optimization approaches is that they are principled, in that we explicitly optimize for what we desire. This makes them more stable and dependable. Q-learning approaches, on the other hand, only indirectly improve agent performance by training Q_θ to meet a self-consistency equation. Since there are so many failure scenarios in this type of learning, it is less stable. When they do work, however, Q-learning approaches have the benefit of being far more sample e□cient than policy optimization techniques since they can reuse data more e□ciently (Russell & Norvig 2016).

There are a number of algorithms that exist on this spectrum and are capable of carefully balancing the strengths and limitations of each side. Sample e□ciency is a problem for some of the most e□ective RL algorithms in recent years, such as trust region policy optimization (TRPO), proximal policy optimization (PPO), and asynchronous actor-critic agents (A3C) (Haarnoja et al. 2018).

2.4.2 MARL Approaches

Lowe et al. (2017) discussed that the Q-learning-based reinforcement learning algorithm is not fit for multi-agent scenarios and policy gradient methods are better for the same. Each agent learns independently optimal Q function but as agents are independently updating their policies as learning progresses, the environment appears non-stationary from any agent's view and violates the Markov assumptions required for convergence of Q learning. It was further discussed that in the policy gradient methods, the policy is generally probabilistic and su□ers from a high degree of variance; multi-agent systems amplify this. The paper proposes using deterministic policies to avoid big variations between episodes. The paper also proposes the use of Actor-Critic networks where training will be centralized, and execution will be decentralized.

Schulman et al. (2017) discussed that Actor-Critic methods are sensitive to perturbations as small changes in underlying parameters of neural networks can lead to large changes in policy space. Proximal Policy Optimization (PPO) addresses this by limiting the updates to policy

networks. The ratio of the new policy to the old policy is constrained to ensure that the gradient step is not huge. The ratio is needed to be in a certain range that is used to create a clipped loss function to avoid very high values of loss.

Zhou et al. (2021) and Chen et al. (2022) proposed policy-based advantage actor-critic algorithms to model lane-changing for connected autonomous vehicles.

There are benefits of policy-based methods over Q-learning methods. The state-of-art papers on lane-changing for AVs/CAVs also used policy-based methods. Therefore, a policy-based RL algorithm, proximal policy optimization (PPO) is used in this dissertation. This is explained in more detail in section 3.1.3.

Partial observability is a fundamental premise of MARL setups as the agents only have access to their local observations rather than the entire state of the environment since the agents are often dispersed throughout the environment.

Some challenges are inherent in MARL settings, like the aforementioned partial observability and non-stationarity of the environment as the environment is a□ected not only by one agent but all the agents present. Therefore, the learning of one agent is unstable in multi-agent RL as the environment is no longer stationary from each agent's perspective. This leads to learning instability due to which agents don't converge to the optimal solutions (Canese et al. 2021). However, agents can communicate information such as observations, intentions or experiences to stabilize learning. The learning is stabilized as the shared experience prevents the network from overfitting to recent experiences and improves sample e□ciency. With communication, agents will better understand the environment (or the other agents), and, therefore can coordinate their behaviours (Zhu et al. 2022).

2.4.3 Cooperative MARL

Generally, MARL algorithms can be classified into *cooperative* and *non-cooperative*. In particular, in the cooperative setting, agents collaborate to optimize a common long-term return and achieve a common goal; while in the competitive setting, the return of agents usually sum up to zero (Zhang, Yang & Başar 2019).

In this paper, the focus is on the cooperative environment, where all agents are encouraged to collaborate in order to accomplish a shared objective, namely, safe maneuvering with maximum throughput.

Generally, in the cooperative MARL setup the training of agents can be performed in three ways: centralized, concurrent and parameter sharing (Gupta et al. 2017).

In the centralized approach, the learning happens jointly for all the agents. The joint observation of all agents is mapped to a joint action using a centralized policy. These approaches

are resource intensive as the state space increases with the number of agents, which hinders scalability.

In contrast to centralized methods, in concurrent learning approaches, several agents learn independently while operating in the same environment. There are individual networks, policies, observations, and actions for each of them. The drawback with this approach is that every agent is learning through a different policy, so the number of parameters is huge and there is no sharing of information. Furthermore, it might lead to instability as the environment is non-stationary because each agent is learning on their own independently of others.

When similar agents are learning similar behaviours, their parameters can be shared to enhance the speed of learning and decrease the complexity and resource utilization of the algorithm (Kaushik et al. 2019). This is possible in the parameter-sharing cooperative approach. Gupta et al. (2017) have discussed that if the agents are homogeneous, their policies may be trained more efficiently using parameter sharing. In the parameter-sharing approach, we allow all the agents to share the parameters of a single policy. This allows the policy to be trained with the experiences of all agents simultaneously. However, it still allows different behaviour between agents because each agent receives different observations, which includes their respective index.

The MARL approach discussed by Zhou et al. (2021) and Chen et al. (2022) uses parameter sharing and experience replay memory. The replay memory or buffer contains the shared experiences of all agents, which is used to update the parameters of the decentralised policy after the completion of one episode. This approach would benefit the simulation of connected autonomous vehicles (CAVs), as this highlights the notion of collaboration and cooperation among agents using a decentralized policy.

2.4.4 Summary

In this section the varying range of RL algorithms for multi-agent RL (MARL) was discussed and the use of model-free policy-based RL for MARL was justified. The techniques to develop cooperative agents was also discussed. The use of PPO algorithm to model lane-changing in CAVs as cooperative-MARL is explained in detail in section 3.2

2.5 Safety approaches in MARL

2.5.1 Overview

Deep reinforcement learning techniques are able to maximize the intended reward, but they may not always ensure safety throughout the learning or execution stages.

This section focuses on studying the implementation of MARL in other domains and exploring

different safety approaches applied there. As inferred in Section 2.3 the safety in lane changing approaches are often limited to introducing safety through the reward function. However, safety in MARL implementation can be introduced through a few other approaches as well, which are discussed below;

Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes. (García & Fernández 2015)

While training the reinforcement learning (RL) agent requires exploring many states to learn optimal policy. In the exploration phase the agent might encounter some unsafe states which makes Reinforcement Learning approach unsuitable for safety-critical systems as here failure can be risky (Thananjeyan et al. 2020) . As a result, safe reinforcement learning is an emerging topic of research that combines control-theoretic techniques with RL to address this problem.

Generally, in the literature, the safety is introduced into RL method using two different approaches. First approach exclusively focuses on getting a safe policy at the conclusion of training (Geibel & Wysotzki 2005, Chow et al. 2017) and second approach focuses on safe exploration during training (Cheng et al. 2019, Berkenkamp et al. 2017).

In this paper the later approach is discussed. García & Fernández (2015) have discussed this approach which involves adjusting the exploratory process to take into account external knowledge (such as teacher advice or demonstrations) or to follow a specific of a risk metric.

2.5.2 External safety supervisors

Most previous papers discussed in section 2.3 focused on optimizing policies based on returns and adding safety constraints to the reward function. But none truly guarantees safety, e.g. no unsafe state is ever visited during the training and execution process.

ElSayed-Aly et al. (2021) proposed to enforce safety specifications as a “shield” using a formal method called Linear Temporal Logic (LTL), a commonly used specification language in formal methods for safety-critical systems. The shield guarantees safety during learning by monitoring the RL agent’s actions and preventing the exploration of any unsafe action that violates the LTL safety specification. Here the exploration process of the RL agent is modified through the incorporation of external knowledge i.e. shields act similarly to a teacher who provides information (e.g., safe actions) to the learner when necessary.

Zhang, Bastani & Kumar (2019) proposed model predictive shielding where the algorithm predicts the next states of each agent when the learnt policy is followed. If the predicted next states are recoverable for all agents it uses the learnt policy, otherwise uses a recovery policy for the agents who will move into irrecoverable states after the current action.

Cai et al. (2021) discussed that relying on the reward functions of MARL only is not sufficient to guarantee safety. It proposes that control barrier functions (CBF) can be used to shield unsafe actions, and CBF lead to minimal interference to the action of an agent. The composite CBF corrects an agent's action only if it violates safety constraints, and the composite CBF revises the action of an agent as few as possible. In the proposed safe MARL framework, an agent has its CBFs that can be different from the CBFs of other agents.

Contrary to Qin et al. (2021), in (Ames et al. 2017) CBF is non-learning where the CBF h is computed first using optimization methods like Sum-of-Squares, then the control inputs u are computed online using h by solving quadratic programming problems.

Choi et al. (2020) presented a RL-framework that formulates quadratic programming incorporating control barrier functions (CBFs) and control laypunov functions (CLFs) as constraints with an objective to minimize the norm of the control input. They proposed to learn the uncertainty in CBFs and CLFs through reinforcement learning and the quadratic program uses the learned uncertainties in combination with safety and stability constraints to solve for the control input point-wise in time.

Emam et al. (2021) framed safety as a differentiable control barrier function layer. They used Gaussian Process to learn the model dynamics which is used by the control barrier function to minimally alter the action from RL algorithm to ensure safety of the agent.

Qin et al. (2021) proposed a framework of jointly learning safe multi-agent control policies and CBF safety certificates. The algorithm defines a combined loss function for CBFs and control policy, during the training the CBFs regulate the control policies to satisfy the decentralized CBF conditions. They proposed a decentralized control framework that scales to an arbitrarily large number (>1000) agents. They use online policy refinement, i.e. the CBF monitors the actions that are input to the controller and CBF refines the action u and enforces the states of agents to remain in a safe set.

Cheng et al. (2019) proposed a framework for combining model-free RL algorithms with control barrier functions (CBFs) to ensure safety and increase exploration effectiveness even in the presence of uncertain model information. They use Gaussian processes to estimate the unknown model dynamics.

Zanon & Gros (2021) suggested a strategy that would integrate Model Predictive Control (MPC) and RL to ensure safety. RL is used to tweak the MPC parameters, enhancing closed-loop performance in a data-driven manner. MPC is utilized as a function approximator within RL to give safety and stability assurances.

Thananjeyan et al. (2020) proposed using a composite policy π which selects between a task-driven policy and a recovery policy at each time step based on whether the agent will violate safety constraints in the near future.

2.5.3 Summary

From the discussion in this section, it can be inferred that safety techniques like shielding, control barrier functions, model predictive control and recovery policies are used in MARL implementation for different domains. In the following sections these approaches are evaluated to understand their functioning, limitations and the possibility of using them for MARL in lane changing for connected autonomous vehicles.

2.6 Conclusion

In Section 2.2 it was observed that reinforcement learning approaches are most widely used to model motion planning in CAVs. Therefore, this dissertation uses multi-agent reinforcement learning to model the lane changing in connected autonomous vehicles.

In Section 2.3 it was observed that most of the implementation of lane changing in CAVs use reward or utility function to ensure safety. However, there is a research gap in usage of external safety techniques for lane changing in CAVs. In Section 2.5 various external safety approaches for MARL were discussed.

This dissertation aims to evaluate these external safety techniques and examine the feasibility of using them in multi-agent RL for connected autonomous vehicle scenarios.

3 Methodology

To answer the research question of whether external safety supervisor can enhance the safety of lane changing in connected autonomous vehicles, we first need to model the lane changing in connected autonomous vehicles as a multi-agent reinforcement problem. Then the external safety approaches discussed in Section 2.5 can be analysed to understand the possibility of integrating these approaches with multi-agent reinforcement learning.

In this chapter I briefly discuss the preliminaries of reinforcement learning (RL) in Section 3.1.1 and multi-agent reinforcement learning (MARL) in Section 3.1.2. After that I discuss in detail how lane changing in connected autonomous vehicles (CAVs) can be modelled as a multi-agent reinforcement problem. In the following section

3.1 Multi-agent RL (MARL) Formulation

This section discusses the background on single-agent (3.1.1), multi-agent reinforcement learning (3.1.2) and a policy based RL algorithm called Proximal policy optimization (3.1.3).

3.1.1 Preliminaries of Reinforcement Learning (RL)

Reinforcement Learning (RL) enables an artificial agent to an learn optimal strategy from the environment through trial and error. Generally an RL problem is modelled as a partially-observable Markov decision process (PO-MDP) because an agent can only observe part of the state s_t. It is called a Markov decision process because it follows the Markov property i.e. the future state of the environment only depends on the present state.

The formal representation of a partially observable MDP is defined as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma)$. An agent at each time step t observes state $s_t \in \mathcal{S} \subseteq \mathbb{R}^n$ and takes an action $a_t \in \mathcal{A} \subseteq \mathbb{R}^m$. Upon taking the action the agent receives reward $r_t \subseteq \mathbb{R}$ based on reward function $\mathcal{R}$, also, subsequently moves to state s_{t+1} at time step t_{t+1}. The transition of agent from s_t to s_{t+1} is generally called one iteration. The transition of agent through a sequence of states ending in terminal state is called an episode.

The RL agent interacts with the environment to learn the optimal policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$, a

mapping from states to actions, that maximizes the expected future reward $R_t = \sum_{k=0}^{T} \gamma^k r_{t+k}$, where r_{t+k} is the reward at time step $t+k$ and $\gamma \subset (0, 1]$ is the discount factor that quantifies the relative importance of future rewards.

The state-value function $V(s)$ returns the expected reward when an agent starts at state s and follows policy π afterwards. The optimal state-value function $V^*(s)$ returns the expected reward when an agent starts at state s and follows the optimal policy π^* afterwards. Similarly, the action-value function $Q(s, a)$ returns the expected return obtained by selecting an action a in state s and following the policy π afterwards. The optimal action-value function $Q^*(s, a)$ returns the expected reward when agent starts at state s, takes an action a and follows the optimal policy π^* afterwards.

In model-free reinforcement learning, the policy π is represented by a neural network parameterised by the learnable parameter θ. The goal is to learn suitable θ values so that optimal agent behaviour is achieved. The optimal behaviour of an agent can be quantified using a reward function.

For any action a_t taken by the agent in any state s_t the reward function is represented in equation 1

$$r_t = R\left(s_t, a_t\right) \tag{1}$$

The policy is optimal when the cumulative rewards over a trajectory is maximum. The return (cumulative reward) can be represented in two ways:

A finite-horizon undiscounted return, which is the sum of rewards after a fixed window of steps:

$$R(\tau) = R \sum_{t=0}^{T} r_t \tag{2}$$

An infinite horizon discounted return, which is the agent's sum of all the rewards discounted by how far in the future each reward was earned. This reward calculation incorporates a discount factor $\gamma \in (0, 1)$.

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t \tag{3}$$

In reinforcement learning sometimes an action can be describe as better than other not through absolute values like in equation 1 but by how much it is better than others on average i.e. relative advantage of an action over other actions in that state. This is represented by an advantage function:

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s, a) - V^{\pi}(s) \tag{4}$$

As discussed in section 2.4 in this dissertation model-free policy based algorithms are used to develop lane-changing algorithm for CAVs. The policy gradient algorithm's core principle is to increase the probabilities of actions that result in higher returns and decrease the probabilities of actions that result in lower returns until you reach the best course of action (Achiam 2018).

Let π_θ denote the policy with parameters θ and $J(\theta)$ denote the expected finite-horizon undiscounted return of the policy. The aim is to maximize the expected return $J(\pi_\theta) = E_{\tau\sim\pi_\theta}[R(\tau)]$.

The gradient of $J(\theta)$ is

$$\nabla_\theta J(\pi_\theta) = \underset{\tau\sim\pi_\theta}{E}\left[\sum_{t=0}^{T} \nabla_\theta log\pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t)\right] \tag{5}$$

where A^{π_θ} is the advantage function for the current policy and τ is a trajectory.

The policy gradient algorithm works by updating policy parameters via stochastic gradient ascent on policy performance.

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta J(\pi_{\theta_k}) \tag{6}$$

where α is the learning rate and $\alpha\nabla_\theta J(\pi_{\theta_k})$ is the change (or step change) in the parameters.

3.1.2 Preliminaries of Multi-agent RL (MARL)

When more than one agents are involved, the MDP is no more suitable for describing the environment as the actions from other agents are strongly tied to the state dynamics (Canese et al. 2021). The multi-agent extension of partially observable Markov decision process (POMDP) is called partially observable Markov games (Littman 1994).

The formal representation of a partially observable Markov game is defined as the tuple $\left(N, \mathcal{S}, \{\mathcal{A}^i\}_{i\in N}, \{\mathcal{R}^i\}_{i\in N}, \gamma\right)$. N is the number of agents. $\mathcal{S}$ is the state-space of the environment, $\mathcal{A}^i$ is the action space of i^{th} agent, $\mathcal{R}^i$ is the reward function of i^{th} agent representing the reward on transitioning from (s_t, a_t) to s_{t+1} and γ is the discount factor for to calculate discounted reward over a series of transitions. In the partially observable Markov game, $\mathcal{O}_i \in \mathcal{S}_i$ is the partial observation of the environment state.

In single agent RL, the agent follows a policy π and aims to optimize the same. Similarly in multi-agent RL, each agent i follows a stochastic policy $\pi_{\theta i} : \mathcal{O}_i x \mathcal{A}_i \rightarrow [0, 1]$.

In multi-agent setting the aim is to maximize the expected return $J(\pi_{\theta i}) = E_{\tau\sim\pi_{\theta i}}[R_i(\tau)]$.

The gradient of $J(\pi_{\theta i})$ is

$$\nabla_{\theta i} J(\pi_{\theta i}) = \underset{\tau\sim\pi_\theta}{E}\left[\sum_{t=0}^{T} \nabla_{\theta i} log\pi_{\theta i}(a_{t,i}|s_{t,i})\, A^{\pi_{\theta i}}(s_{t,i}, a_{t,i})\right] \tag{7}$$

where $A^{\pi_{\theta i}}$ is the advantage function for the current policy and τ is a trajectory.

The policy gradient algorithm updates the policy parameters iteratively for each agent i via stochastic gradient ascent on policy performance.

$$\theta_{ik+1} = \theta_{ik} + \alpha\nabla_{\theta i} J(\pi_{\theta_i k}) \tag{8}$$

where α is the learning rate and $\alpha\nabla_\theta J(\pi_{\theta_k})$ is the change (or step change) in the parameters.

3.1.3 Preliminaries of Proximal Policy Optimization for MARL

The proximal policy optimization (PPO) algorithm aims to take largest step to improve the policy while ensuring that there is not a very large di□erence between the old and new policy which might result in poor performance. It relies on specific clipping in the objective function to eliminate incentives for the new policy to diverge much from the previous one. Proximal policy optimization (PPO) is more sample e□cient in learning policies which means it requires less data to reach better performance (Ye et al. 2020) .

PPO updates policies via

$$\theta_{k+1} = \arg\max_\theta \underset{s,a\sim\pi_{\theta_k}}{\mathrm{E}}\left[L(s, a, \theta_k, \theta)\right], \tag{9}$$

typically taking multiple steps of (usually minibatch) SGD to maximize the objective. Here L is given by

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a),\ \ \mathrm{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_k}}(s, a)\right), \tag{10}$$

in which ϵ is a (small) hyperparameter which roughly says how far away the new policy is allowed to go from the old.

Clipping acts as a regularizer by eliminating incentives for the policy to change significantly, and the hyperparameter ϵ measures how far the new policy can diverge from the old without harming the goal (Achiam 2018).

In this dissertation a PPO based cooperative multi-agent reinforcement algorithm is developed to model lane changing in connected autonomous vehicles. The model is explained in detail in next section.

3.2 Lane changing in Connected Autonomous Vehicles as MARL (MARL-CAV)

In this section, a decentralized MARL-based approach for highway lane-changing of multiple CAVs is discussed. This implementation is a customized version of MARL implementation in (Chen et al. 2022).

This implementation is adapted from (Chen et al. 2022). They have also used the same simulation environment, however their problem statement was di□erent. They worked on developing MARL techniques for ramp-merging scenario. I have followed their implementation open-sourced in Github repository. I modified it for lane changing environment and added a custom reward function discussed below. The specifics of the modification is discussed in more detail in Section 5.

The mixed-tra□c lane-changing environment is modelled as a multi-agent network: $\mathcal{G} = \{V, \varepsilon\}$, where each agent $i \in v$ communicates with neighbours $\mathcal{N}_i$ using the communication link $\epsilon_{ij} \in \varepsilon$. The overall dynamic system can be considered a partially-observable Markov decision process (POMDP) which can be represented by the tuple $(\{\mathcal{A}_i, \mathcal{S}_i, \mathcal{R}_i\}_{i \subseteq v}, \mathcal{T})$, where $\mathcal{A}_i$ is the local action space, $\mathcal{R}_i$ is the reward space, $\mathcal{O}_i \in \mathcal{S}_i$ is the partial observation of the environment state (Chu et al. 2020, Zhou et al. 2021).

In partially observable Markov games (multi-agent POMDP), every agent follows a decentralized policy $\pi_i : \mathcal{O}_i x \mathcal{A}_i \rightarrow [0, 1]$ to chose its own action $a_{i,t} \sim \pi_i(\cdot|s_{i,t})$ at time step t. Following is the description of individual components of the discussed POMDP setting:

1. Action Space: The action space $\mathcal{A}_i$ of agent i is defined as a set of high-level control decisions, including: cruising, turn left, turn right, speed up and slow down. The action space combination for CAVs is defined as $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$, where N is the total number of vehicles in the environment. After receiving the high-level action, the low-level controllers will produce corresponding throttle and steering signals to maneuver the CAV.
2. State Space: The state space $\mathcal{O}_i$ of Agent i is defined as a matrix of dimension $\mathcal{N}_{N_i} \times \mathcal{F}$ where $\mathcal{N}_{N_i}$ is the number of detected vehicles, and $\mathcal{F}$ is the number of features, which is used to represent the current state of vehicles. It includes the following features:
    - x: the longitudinal position of the observed vehicle relative to the ego vehicle

- y: the lateral position of the observed vehicle relative to the ego vehicle
- v_x: the longitudinal speed of the observed vehicle relative to the ego vehicle.
- v_y: the lateral speed of the observed vehicle relative to the ego vehicle.

3. Reward Function: The design of the reward function is important to ensure that the agent achieves the goal of safe lane changing. In the proposed work, the reward function is composedly designed using multiple metric like safety, headway distance, driving speed and right lane driving.

$$r_{i,t} = w_s r_s + w_h r_h + w_d r_d + w_{rl} r_{rl} + w_{lc} r_{lc} \tag{11}$$

where $w's$ are weighing coe□cients and $r's$ are cost evaluation. Following are the details:

Table 3.1: Reward function components

Component	Description
w_s	Penalty on collision
w_h	Incentive to maintain a good headway distance i.e., distance to front vehicle
w_d	Incentive to maintain a stable speed, so that the agent does not slow down in order to always avoid collision
w_{rl}	Incentive to follow right lane driving. It in-turn encourages the vehicles to change lanes, which start on left or center lens.
w_{lc}	Incentive to change lanes.

.

In the proposed approach a deep neural network is used to approximate the stochastic decentralized policy π of the RL agents. This network is shared between all agents, apart from this, a shared replay bu□er is also maintained that stores the experiences from all agents. A copy of state information i.e. observations, actions and rewards is held by the individual agents. No agent has access to the state information of any other agent. However, as the data in Replay Bu□er is shared and identical, each agent benefits from the collective experiences of all agents. Finally, each agent updates the policy network asynchronously at each step (Kaushik et al. 2019). The process flow of the multi-agent RL is shown in Figure 3.1

The pseudo-code of the proposed MARL algorithm is shown in algorithm 1. The algorithm

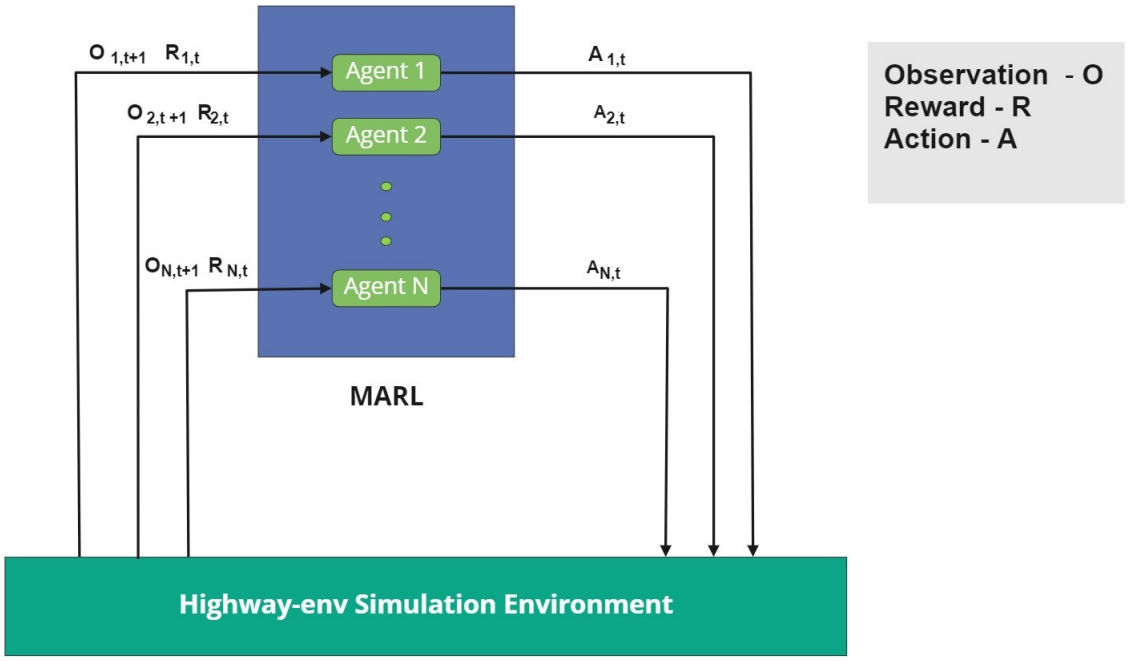


Figure 3.1: Flow chart of one episode where agents perform action $A_{n,t}$ and receive reward $R_{n,t}$ and observe the next state $O_{n,t+1}$

is adapted from Zhou et al. (2021), Chen et al. (2022) and is very much similar to their approach of using parameter sharing and replay bu□er to enable cooperative behaviour and collaboration among multiple connected agents/vehicles. The reasoning behind parameter sharing and replay bu□er was discussed in Literature Review section 2.4.

The hyperparameters include: the (time)-discount factor γ, the learning rate α, the total number of training episodes M, the episode length T. The agent receives the observation s_i from the environment and updates the action depending on its policy (Lines 6-9). The action will be taken by the agent and the corresponding experience will be collected and saved to the replay bu□er (Lines 10-13). After each episode is completed, the network parameters are updated accordingly (Lines 9-11). The parameters of the policy network are updated using the collected experience sampled from the on-policy experience bu□er after the completion of each episode (Lines 15–16). The DONE signal is flagged if either the episode is completed or a collision occurs. After receiving the DONE flag, all agents are reset to their initial states to start a new episode (Line 23).

Algorithm 1 MARL for CAVs

Input parameters: γ, α, T, M
Output: θ
Initialize: $o_i, j \leftarrow 0, t \leftarrow 0, \mathcal{D} \leftarrow 0$
for $j \leq M$ do
    for $t \leq T$ do
        for $i \in V$ do
            observe s_i
            update $a_{i,t} \sim \pi_i(\cdot|s_{i,t})$
        end for
        for $i \in V$ do
            execute a_i
            update $\mathcal{D}_i \leftarrow (s_{i,t}, a_{i,t}, r_{i,t}, v_{i,t})$
        end for
        update $t \leftarrow t + 1$
        if *DONE* then
            for $i \in V$ do
                update $\theta_i \leftarrow \theta_i + \alpha \nabla_{\theta_i} J(\theta_i)$
            end for
        end if
        Initialize $\mathcal{D}_i \leftarrow 0, i \in V$
        update $j \leftarrow j + 1$
    end for
    Initialize: $o_i, j \leftarrow 0, t \leftarrow 0$
end for

3.3 Summary

This chapter discussed the overall methodology to answer the research question. In particular, the reasoning behind the use of multi-agent RL to model the lane changing in connected autonomous vehicles was discussed. The Proximal policy optimization (PPO) algorithm used for training the MARL for lane changing was discussed. Also, the mathematical formulation, action space, state space and reward function of the proposed MARL-CAV model was extensively discussed. It was further highlighted how this was adapted and modified for the lane-changing use case in this dissertation.

Finally, the MARL algorithm for lane changing in CAVs was extensively discussed. The following chapter discusses safety supervisor approaches and their applicability.

4 Safe MARL-CAV Design

In Section 2.5 I discussed various formal methods used to introduce an external safety supervisor for an agent in reinforcement learning algorithm, especially in a multi-agent setting. This chapter discusses the safety requirements of MARL-CAV in Section 4.1 and then presents a detailed analysis of these methods in Section 4.2 to verify the possibility of using these approaches to introduce external safety in MARL-CAV.

4.1 Safety Requirements for lane changing in CAV

The design of MARL-CAV discussed in Section 3.2 only uses the reward function to introduce some level of safety. This is achieved by penalising actions which lead to a collision and rewarding the agents when they maintain proper headway distance from the front vehicle. However, this is not enough to stop the agent from visiting unsafe states during the exploration phase. Hence, an external safety supervisor might be useful to stop the agent from visiting unsafe states.

An external safety supervisor can be applied to enhance the agent's safety. The safety approaches discussed in Section 2.5 namely shielding, control barrier functions, model predictive control, and recovery RL, should integrate with MARL-CAV and then we can examine if these techniques can improve safety.

However, the safety approach should be compatible with the characteristics of the MARL-CAV model. It should be simple, extensible and adaptable enough to easily integrate with the MARL in lane changing for connected autonomous vehicles.

The table 4.1 shows the requirements specific to the MARL-CAV that the safety approach should satisfy. The safety technique should support multiple controlled agents and the presence of uncontrolled agents e.g. in lane-changing for CAVs, human-driven vehicles are also present.

Also, one requirement specific to the simulation library highway-env (Leurent 2018) is that the safety approach should be compatible with discrete action space as this library only supports discrete actions: move to the left lane, move to the right lane, forward, idle.

Table 4.1: Characteristics of Lane Changing in CAVs

Characteristic	Requirement for Safety Approach
Number of agents	Should support a multi-agent scenario. It should be scalable to a large number of agents.
Action Space	Should support discrete actions as the current implementation of MARL-CAV has discrete actions.
Unsafe states information	There is no prior information about unsafe states as the simulation is very dynamic.
Recovery/Backup Policies	There is no recovery or backup policy developed to get the agent from unsafe to safe states.
Mixed traffic scenario	Should support the presence of both controlled and uncontrolled agents in the environment.

Furthermore, in the current implementation of MARL-CAV, the environment is quite dynamic, and there is no prior information about recovery policies or unsafe state information.

If it does not satisfy, it would require modification of either the MARL-CAV set-up or tweaks in the safety approach. This would make the process of introducing external safety in MARL-CAV more challenging.

The next section analyses the applicability of the safety techniques to MARL-CAVs and examines if they satisfy the requirements mentioned in the table 4.1.

4.2 Analysis of External Safety Techniques for MARL

In section 2.5, several external safety techniques applied for reinforcement learning were discussed. Generally, control-theoretic or formal methods like shielding, control barrier functions, model predictive control and recovery policies are used to introduce external safety in MARL implementation for different domains. These methods are reviewed in more detail in this section, and their benefits and limitations are discussed.

4.2.1 Shielding

Shields are modelled for an RL agent using finite state machines or, more specifically, using Mealy machines. A linear temporal logic (LTL) safe specification can be translated into a safe language accepted by deterministic finite automaton (DFA) (Kupferman & Vardi 2001). In deterministic finite automaton, a Mealy machines is represented using the tuple $(Q, q_0, \Sigma_O, \Sigma_I, \delta, \lambda)$ with a finite set of states Q, initial state $q_0 \in Q$, finite sets of input alphabet Σ_I and output alphabet Σ_O, the transition function $\delta : Q x \Sigma_I \rightarrow Q$, and the output

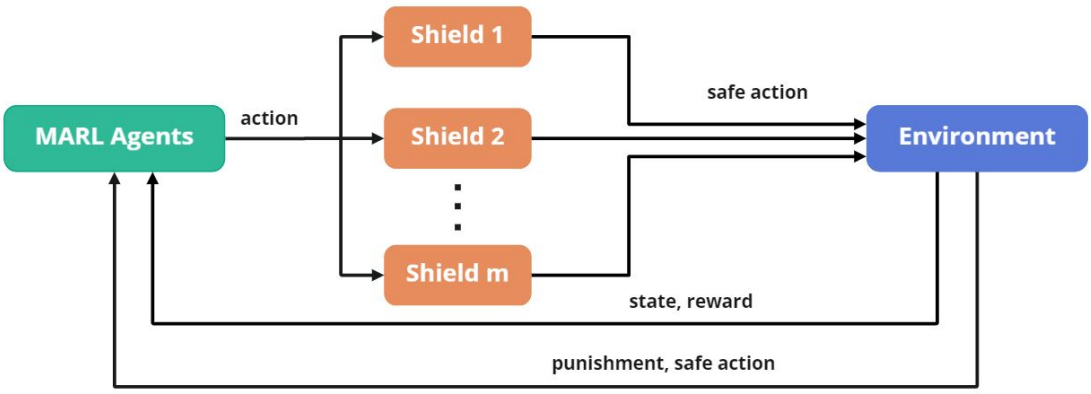


Figure 4.1: Diagram represents the functioning of the shielding technique (ElSayed-Aly et al. 2021)

function $\lambda : Q x \Sigma_I \rightarrow \Sigma_O$. The shields were synthesized using the Slugs tool (Ehlers & Raman 2016) via solving two-player safety games. More details on this can be found in the paper (ElSayed-Aly et al. 2021).

Figure 4.1 shows the shielding technique in a multi-agent setup. As discussed in (ElSayed-Aly et al. 2021), here shields are used to update the action of RL agents to ensure that agents stay in safe states.

ElSayed-Aly et al. (2021) claim that the shielding approach not only guarantees safety but also learns more optimal policies with better returns than non-shield MARL, as unsafe actions which may destabilize learning were removed.

However, shielding requires prior knowledge of safe states in the environment so that they can be used to design shields. Safety is specified using Linear Temporal Logic (LTL). This might be possible for a simpler environment like navigation tasks in ElSayed-Aly et al. (2021) but could be di□cult for complex environments like lane changing in connected autonomous vehicles. As discussed in the Table 4.1, there is no prior information about unsafe states in the MARL-CAV implementation.

4.2.2 Control Barrier Functions (CBFs)

Control Barrier Functions (CBF) is a model-based safety framework that prevents the exploration of dangerous states by projecting the RL agent's actions onto a safe set of actions.

As discussed Grandia et al. (2021), with CBFs the idea of safety is specified by defining a safe set in the state space in which the system is required to stay. Given a time-varying set $C \subset \mathbb{R}^n$ defined as zero superlevel set of a continuously di□erentiable function $h : \mathbb{R}^n x \mathbb{R}_+ \rightarrow \mathbb{R}$

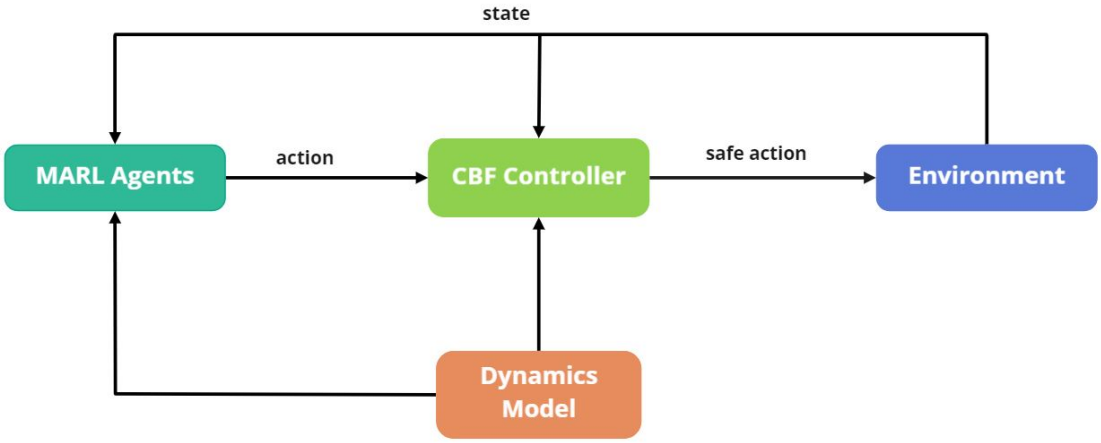


Figure 4.2: Diagram represents the functioning of CBFs (Emam et al. 2021)

$$C \triangleq \{x \in R : h(x,t) \geq 0\} \tag{1}$$

where C is the safe set.

As discussed in Section 2.5, several approaches to MARL have introduced external safety through CBFs. These papers have simulated di□erent tasks such as navigation tasks, i.e. agent aims to move from a source to destination like in (Qin et al. 2021, Cai et al. 2021), or any OpenAI (Brockman et al. 2016) tasks like inverted pendulum in (Cheng et al. 2019), car following environment (Cheng et al. 2019, Emam et al. 2021), unicycle environment (Emam et al. 2021) and bi-pedal walker (Choi et al. 2020).

In these works authors have considered those tasks as a dynamic control a□ne system. Control a□ne system as these are non-linear in control input u.

$$\dot{x} = f(x) + g(x)u(x) \tag{2}$$

where $x \in C \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^m$. Here C is set of all possible safe states of the system, and U is a set of all admissible control inputs. Furthermore, $f(x) \in \mathbb{R}^n$ is drift dynamics and $g(x) \in \mathbb{R}^{nxm}$ is the input dynamics. $u(x)$ is the control input which is the output of the reinforcement learning algorithm. The assumption is that $u(x)$, $f(x)$ and $g(x)$ are continuous functions.

A time-varying set C is safe if for every $x_0 \in C_0$, the solution x_t to (2) satisfies $x_t \in C_t$ for all $t \in [0, t_{max})$. The system in Equation 2) is safe on the set C_t if the set C_t is forward invariant and satisfies Equation 1.

The idea of forward invariance is that if we have a state, we want to make sure that the state of the system would stay inside the set for long time. Control Barrier Functions can be

used as a formulation tool to achieve forward invariance and, therefore the safety of a set. This is a very promising idea and hence control barrier function is the most widely used safety technique out of the four techniques discussed.

As discussed in Section 2.5, Qin et al. (2021) developed a joint framework to learn CBF safety certificates and multi-agent control policies. However, they are assuming RL as MDP and have state information S_i available during the RL agent training process, whereas as discussed in Section 3.2, the lane changing in CAVs using MARL is a partially observable MDP (POMDPS). Hence S_i wouldn't be available. Also, they have considered independent policies for each agent but in our setup, as discussed in Section 3.2 we have the same policy with shared parameters.

Cheng et al. (2019) assume knowledge of CBF and model dynamics. During training, Gaussian Processes (GPs) are used to learn the dynamics' uncertainty, which is then accounted for in the CBF-based safety layer. They used a continuous action space in both the inverted pendulum and car following tasks. Furthermore, they have only considered environments with controlled agents but not a mix of controlled and uncontrolled agents, which will be in case of lane changing scenario, as it will have both controlled vehicles and human-driven vehicles.

Similar to the approach of (Cheng et al. 2019), Emam et al. (2021) also used the Gaussian process to model the system dynamics, which was used by CBF controller to learn about unsafe states and also used by RL agents to improve sample e□ciency (training with less data). The action of the RL agents is updated by the CBF controller. Figure 4.2 represents the function of CBF implementation in (Emam et al. 2021).

These approaches (Zhao et al. 2021, Cheng et al. 2019, Qin et al. 2021) used continuous action space as CBF would not work for a discrete action space. Control barrier functions do not work for discrete action space because with discrete action space the RL model cannot be represented using di□erential dynamic programming and the lie derivatives would not work. This is a major drawback as per the safety requirements discussion in Section 4.1

A key problem for this approach is figuring out how to combine knowledge of model (environment) dynamics with model-based safety. Because control barrier functions are model-based i.e. they require information about model dynamics. Hence most of these papers either use model-based RL like in (Emam et al. 2021) or use a statistical model to learn model dynamics like in Qin et al. (2021), Zhao et al. (2021), Cheng et al. (2019). However, in POMDP discussed in section 3.2 the system dynamics can be very uncertain. There is no guarantee that a complex setting of multi-agent RL for lane changing with mixed tra□c scenario can be described with some assumptions of dynamics.

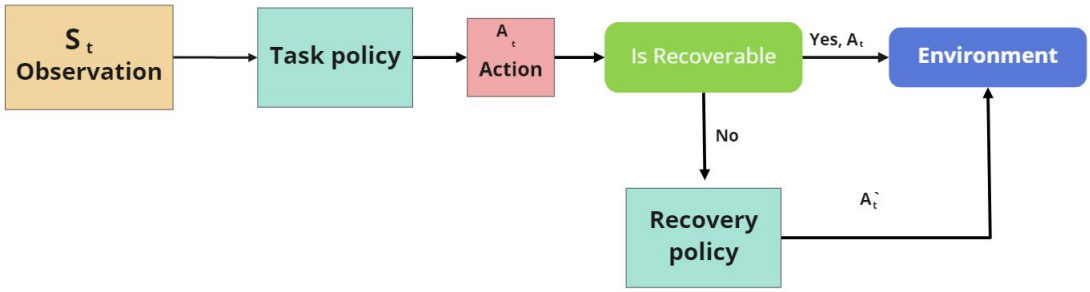


Figure 4.3: Diagram represents the functioning of MPC (Zhang, Bastani & Kumar 2019)

4.2.3 Model Predictive Control (MPC)

Model predictive control can be defined as single or multi step ahead estimation of states that can be reached by an agent over multiple next time steps under a sequence of control inputs. Generally, statistical models are used to approximate the system dynamics, which helps to predict the successive states of the environment when an action is taken by the agent. The predicted states are then verified if they violate safety constraints and if an agent has a high probability of reaching unsafe states, it uses a recovery (or backup) policy that brings it back to safe states.

Model Predictive Control (MPC) methods guarantee the availability of return trajectories to a safe state at every time step with high probability. Koller et al. (2018) proposed a learning-based MPC scheme that provides high probability safety guarantees and helps RL agent learn safer and optimal policies. They used OpenAI inverted pendulum environment (Brockman et al. 2016) to run experiments. They used Ellipsoids (Kurzhanski & Varaiya 2000) to compute reachability states i.e. the predicted future states for the agent and apply the best possible action to bring the agent back to safety.

A joint framework can be used to improve the MPC's safety constraints and RL algorithm's task policy (Zanon & Gros 2021). In this paper, reinforcement learning was used to update the parameters of MPC, and MPC was used to provide a safety guarantee to the RL agent. This is an intuitive way of using closed-loop performance improvement. however, it might lead to oscillatory behaviour, and neither the RL may learn optimal policy nor the MPC parameters are good enough to ensure safety.

Unlike the single-agent RL with MPC approach in (Koller et al. 2018, Zanon & Gros 2021), MPC can also be applied in a multi-agent setting, like in (Zhang, Bastani & Kumar 2019). Figure 4.3 shows the functioning of this approach. In their proposed method, they are incrementally checking whether each agent is in a set of stable states χ_{stable}, and if any agent is predicted to be going into an irrecoverable or unsafe state, then recovery policy $\pi_{recovery}$ is used to bring the agent back to safe states χ_{stable}. The agents predicted to be in safe set after

current action use the respective learned policy π^i. Unlike our implementation of parameter sharing discussed in section 2.4 and 3.2 they have assumed di□erent learned policy for each agent which is not scalable to higher number of agents.

This approach looks promising and able to act as a safety supervisor for the RL agent. However, these approaches generally have two underlying assumptions. First, availability of a recovery policy, also called safe policy π_{safe} is used to return the agent to safe state when it is about to visit the unsafe states. Second, some prior information of either the irrecoverable (unsafe) states or safe states.

The challenge is that the recovery policy and data of unsafe states for an environment might not be available for all use cases. For example, in a multi-agent lane changing the setting, it is di□cult to know which all states are unsafe and how one trains a recovery policy. These challenges are limitations to using this safety technique for multi-agent lane-changing scenarios.

Though CBF has some limitations, but it has been used in a lot of publications and might be the most promising approach for safety in multi-agent RL (MARL).

4.2.4 Recovery RL

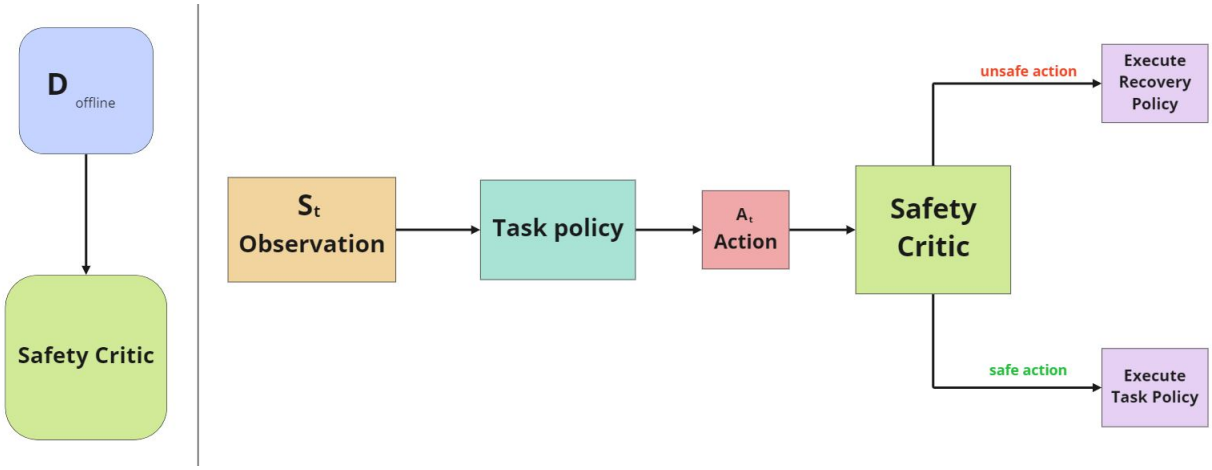


Figure 4.4: Diagram represents the functioning of Recovery RL (Thananjeyan et al. 2020)

The Recovery RL approach is relatively very recent compared to the previous three approaches. Thananjeyan et al. (2020) proposed a unique approach to create an external supervisor for RL algorithms. However, it has some similarities to model predictive control methods, especially the one in (Zhang, Bastani & Kumar 2019).

Figure 4.4 shows the high-level functioning of the Recovery RL technique. An o□ine dataset $D_{offline}$ was generated which contains set of trajectories where the agent violated safety constraints. This was either generated manually using human knowledge or through a naive RL policy. This $D_{offline}$ data was used to train a reinforcement learning-based safety-critic policy. The safety critic was later used to estimate the agent's probability of future constraint violations. If the safety-critic predicts the agent's action to be unsafe, then recovery policy $\pi_{recovery}$ is used to bring it back to safe states or else a learn policy π_{task} is used.

This approach requires us to design the $D_{offline}$ either manually using human knowledge or through a separate reinforcement learning policy. For complex systems like the multi-agent RL model for lane changing in CAVs, it would require the first training an RL policy and extracting the $D_{offline}$. Though the $D_{offline}$ would be updated during the training of the recovery RL it does require to run the RL model without the safety critic to collect $D_{offline}$. This in-turn defeats the purpose of designing an external safety supervisor, as the main aim of external safety is to stop the agent from exploring unsafe states.

4.2.5 Summary

Table 4.2: Analysis of the External Safety Supervisors for MARL in terms of the scenario requirements. Green color cell means scenario requirement is satisfied

Characteristic	Shielding	Control Barrier Functions	Model Predictive Control	Recovery RL
Number of agents	Supports multi-agent	Supports multi-agent	Supports multi-agent	Supports multi-agent
Action Space	Continuous Action	Continuous Action	Continuous Action	Continuous and Discrete Action
Model dynamics information	Required apriori	Modelled using Gaussian Processes or Assumed	Modelled using Gaussian Processes, or Assumed	Learned using RL
Unsafe states information	Required	Not required	Required	Required
Recovery/Backup Policies	Not required	Not required	Required	Required
Mixed agents (controlled & uncontrolled)	No support yet	No support yet	No support yet	No support yet

Table 4.2 summarises the analysis of safety techniques concerning the safety requirements discussed in Section 4.1. The green color cell means it the safety method satisfies safety requirements. From the table, it can be inferred that none of the safety techniques satisfies all the safety requirements; hence, there are still some open challenges in using them for MARL-CAV. These challenges are discussed in more detail in the next section.

4.3 Open challenges in safety for CAV-MARL

As discussed in Section 2.3, most MARL methods for CAV modelling mostly focus on achieving optimal policies based on their reward function but this does not guarantee safety, i.e., that no unsafe state is ever visited during the learning process. This is because penalizing agents through negative rewards doesn't stop them from going to unsafe states as they have to explore different (both safe and unsafe) actions to learn about the states and rewards from the environment.

The approaches discussed in Section 4.2, namely shielding, model predictive control, control barrier functions and recovery RL, are ways to guarantee safety during the exploration process as they act like an external supervisor, thereby monitoring the agents' actions and preventing the exploration of unsafe states. These approaches have been used to provide external safety for MARL for other scenarios and might be applicable to the multi-agent reinforcement learning algorithms for lane changing in CAVs. The multi-agent proximal policy optimization algorithm discussed in Section 3.1.3 has to be modified to include one of these safety approaches.

However, this implementation will be a significant challenge. Based on the data in table 4.2 it can be inferred that most of these approaches are applied to simpler environment tasks, like one or more agents trying to reach a goal (Qin et al. 2021, Cai et al. 2021) , navigation tasks (Thananjeyan et al. 2020) or simple OpenAI gym environments (Cheng et al. 2019, Emam et al. 2021, Choi et al. 2020). These approaches use statistical models to estimate the dynamics of the system, but this might not work for a complex scenario like lane-changing in CAVs, where other human-driven vehicles, i.e., uncontrolled agents, are present. Compared to the static obstacles present in simpler environments, these represent dynamic obstacles.

Furthermore, as discussed in Section 4.2, these approaches come with several assumptions like the following:

- Availability of partial or complete prior knowledge of unsafe or irrecoverable states in shielding, model predictive control and recovery RL.
- Availability of recovery policies in case of model predictive control and recovery RL.
- Bounded system dynamics for control-theoretic methods like control barrier functions.
- Having only a single or more controlled agents in the system. Absence of uncontrolled agents (like human-driven vehicles in the lane-changing scenarios).

As per the safety requirements in section 4.1, these assumptions will be violated in the MARL-CAV setting discussed in section 3.2 and this posses a significant challenge in adapting these approaches to MARL-CAV. Based on the assumptions and limitations discussed in this chapter, the potential introduction of these safety approaches to MARL-CAV comes with theoretical

and practical challenges.

4.3.1 Conclusion

In this chapter, the safety requirements of MARL-CAV were discussed and four different external safety supervision techniques were analysed to examine their compatibility with MARL-CAV setup. Although all the approaches lack some safety requirement, we need to select a safety technique and integrate it with MARL-CAV to verify whether these techniques enhance safety. Based on the Table 4.2 the control barrier function approach looks simpler as it does not require the availability of recovery policies prior information about unsafe states. Therefore, I decided to implement safe MARL-CAV with control barrier functions (CBF).

Given the limited time, only the CBF approach was tested with MARL-CAV;The implementation of the other three approaches i.e., shielding, model predictive control and recovery RL for MARL-CAV is beyond the scope of this dissertation. However, this can be done in future work.

In this chapter, I discussed the potential theoretical limitations in these safety approaches, however, to answer the research question, this needs to be practically tested. Therefore, the practical aspect is discussed in the following 5 and 6 chapters.

5 Safe MARL-CAV Implementation

This chapter presents the implementation details of MARL-CAV discussed in Section 3.2. Along with simulation details, the challenges, the modification in the simulation environment and set-up details are discussed in di□erent sections.

5.1 Simulation Environment

Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) techniques for lane changing in connected autonomous vehicles are mostly performed on tra□c simulators as it can be unsafe to perform real-life experiments with vehicles. Microscopic tra□c simulators are commonly used to control individual vehicles in the simulation. One of the most popular open-source tra□c simulators is Simulation Urban Mobility (SUMO) (Lopez et al. 2018). However, the tra□c control interface (TraCI) package is required to control a vehicle in SUMO. TraCi enables users to communicate with the SUMO environment using Python. Liao et al. (2021) and Ye et al. (2020) are discussed in the section 2.2 used SUMO for simulation.

Wu et al. (2022) proposed a framework called "Flow" that enables the use of deep reinforcement learning algorithms and perform control experiments for tra□c micro-simulation. I started with this framework, but it was computationally expensive and took hours to simulate a small lane-changing scenario. Moreover, it required managing the configurations of both SUMO and flow framework. Furthermore, flow Wu et al. (2022) is not maintained as seen in its Github page. Due to the above-mentioned issues, I could not use Flow-SUMO and decided to try more actively maintained and minimalist frameworks.

Chen et al. (2022) have used a more pythonic and minimalist environment highway-env Leurent (2018) to simulate connected autonomous vehicles. However, this environment doesn't support the simulation of multiple autonomous and connected autonomous vehicles, but it is possible to extend and customize the environment to achieve the same. Taking inspiration from, Chen et al. (2022) I have used the same environment and modified it to implement a multi-agent reinforcement learning setting for lane changing in connected autonomous vehicles. he modifications are discussed in detail in the next section. The highway-env environment assumes connection between all controlled vehicles. All vehicles receive data of five surround-

ing vehicles as observation or partial state information.

5.2 Simulation Challenges

The highway-env Leurent (2018) library was used for simulation and from the library, the fast-highway-v0 was used for the simulation of lane changing in connected autonomous vehicles (CAVs). Figure 5.1 shows a screenshot of the simulation.

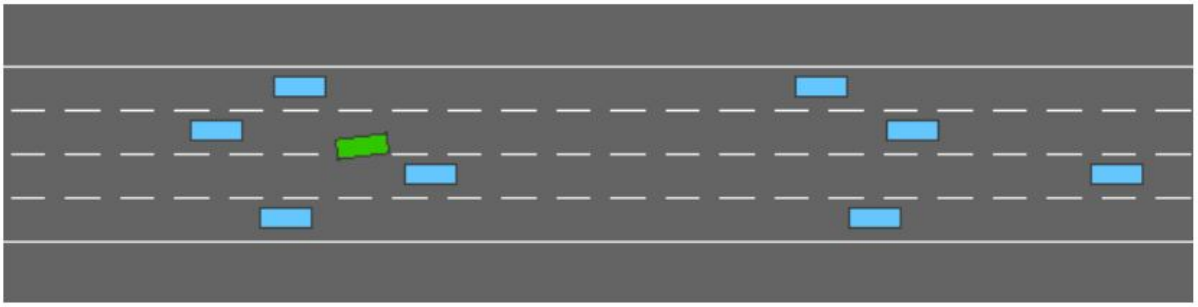


Figure 5.1: Screenshot of the fast-highway-v0 environment with CAVs (in green) and human-driven vehicles (in blue). In this figure the CAV is trying to change the lane to avoid collision with HDV.

However, this library came with its fair share of limitations, as it was not tailor-made to simulate the MARL-CAV discussed in Section 3.2. Following are some of the limitations:

- Doesn't come with support for multi-agent in fast-highway-v0 environment.
- Doesn't come with support for connected autonomous vehicles (reference discussion).
- Doesn't come with support for continuous action.

To overcome these limitations significant amount of time was spent to modifying the source code of the library to enable it to support multi-agent reinforcement learning and connected autonomous vehicles.

First, the highway-env library was modified to support multi-agent RL and registered a new environment fast-highway-MA-v0, which allows simulation of multiple controlled vehicles. Second, the vanilla PPO implementation was modified to include parameter sharing and replay buffer. This enabled the implementation of connected autonomous vehicles as discussed in Section 2.4.3. Furthermore, the reward function was also modified as discussed in the Section 3.2. Following sections talk about the modifications in more detail:

5.2.1 Added support for multi-agent

The highway-env only supports single-agent learning in its environments. To add the support for multi-agent, the reward function was updated to collect rewards for individual agents,

and then individual rewards were combined to receive the collective reward. Furthermore, the proximal policy optimization algorithm was updated to add support for multi-agent training and developed the multi-agent PPO (MA-PPO) algorithm.

5.2.2 Added support for connected autonomous vehicles

The highway-env library does not support connected autonomous vehicles. To make the MARL-CAV implementation support collaboration among different agents, the MA-PPO algorithm was updated to support parameter sharing and add replay buffer functionality discussed in Section 2.4.3. This ensured that agents cooperatively take action.

5.2.3 Added support for continuous action

The highway-env library out-of-box just supported discrete action for fast-highway-MA-v0 environment. Therefore, MARL-CAV was developed to work with discrete action space. These actions were move left, move right, forward and idle.

However, as discussed in 4.3.1 the safety technique, control barrier function (CBF) was selected to be examined whether it adds safety to MARL-CAV. CBFs only support continuous action space. Therefore, the code for fast-highway-MA-v0 environment and MA-PPO algorithm was updated, and support for continuous actions was added. The loss function of MA-PPO was updated to use a Gaussian policy rather than the initial categorical policy. The continuous actions were steering angle and throttle value for the vehicle.

The continuous action PPO setup was tested on Open AI inverted pendulum environment Brockman et al. (2016). The algorithm worked successfully and kept the pendulum in an upright position.

Furthermore, one more component was added to the reward function. This was added to ensure that the vehicle gets rewarded for staying in the centre of the lane while moving in the road.

5.3 Simulation Set-up

The developed environment, i.e., fast-highway-MA-v0, was used to simulate the lane-changing scenario. In a MARL setting, the agent trains for a specific number of episodes, and the reward is calculated at the end of every episode. The calculated reward value can be stored to compare the progress of the agent's learning. Ideally, the reward should increase as the number of training episodes increases.

An episode in a reinforcement learning setting is one simulation of the task given to the agent. In our case, it will be the task of the multiple vehicles (agents) to move from the start of the

road to the end and safely change lanes. An episode terminates when any agents collide or the maximum length is reached.

To run the experiments consistently, I have run the experiments with the following default settings:

Table 5.1: Default simulation settings for MARL-CAV

Setting	Value
Number of CAVs	2
Number of HDVs	6
Number of training episodes	1000
Evaluation interval	200
Number of evaluation episodes	3
Max length of each episode	20 Seconds
Termination settings	Collision or Episode max length reached.

Table 5.1 consists of the default setting for the experiments discussed in the next section. This settings can be tweaked to change the difficulty of learning a good RL policy. For example, if the number of CAVs are increased then it becomes difficult to learn a good policy where all agents behave optimally. In the next section, the standard settings have been used.

6 Evaluation

The intention of implementing the MARL-CAV in Section 3.2 was to verify whether the external supervisor approaches discussed in Section 2.5 can be integrated into the MARL-CAV and to check if external safety supervisor approaches work with MARL-CAV to make it safer.

As discussed in the conclusion of 4.3, control barrier functions (CBF) is the most promising safety technique considering the safety requirements of MARL-CAV (discussed in Section 4.1). The necessary modification required to implement CBF with MARL-CAV is discussed in the previous section 5.2.3. This chapter discusses the results (6.2) of the integration of control barrier functions with MARL-CAV.

6.0.1 Evaluation Metrics

This section discusses the evaluation metrics that can be used to judge the performance of the proximal policy optimization (PPO) algorithm for the MARL-CAV task. This would also help benchmark this algorithm's performance with and without integration of external safety approach CBF.

An RL agent's evaluation is done based on the reward it gets from the environment. When an RL agent follows an optimal policy, the reward is highest. The RL agent receives a reward for each episode, and during training, multiple episodes are simulated to help the RL agent reach optimal policy.

For the evaluation, I have trained the PPO algorithm for 1000 episodes on the MARL-CAV model. After every 200 episodes, I run a couple of evaluation episodes and calculate the metric values.

Here episodic rewards have been used to design the metrics.

The following are two metrics which I implemented for the evaluation:

1. Average reward per episode - This metric gets the mean reward of the evaluation episodes. Ideally, this score should increase over the training episodes to show that the agent is getting better and receiving more rewards.

2. Average length of episode - This metric gets the mean length of evaluation episodes in seconds. Generally, if the agent doesn't follow an optimal policy, it will lead to more collisions, and collisions terminate the episodes. Hence, ideally, as the agent learns optimal policy, the lengths of episodes should be longer.

6.0.2 Evaluation Design

The evaluation of safe MARL-CAV is performed with the help of two experiments. The first experiment in Section 6.1validates the implementation of MARL-CAV i.e., to check whether the implementation is able to learn a stable policy that makes safe lane changes. The second experiment in Section 6.2 examines the applicability of CBF with MALR-CAV.

However, in the safety analysis table 4.2 we can observe that the CBF doesn't support discrete action. Therefore, I decided to convert the MALR-CAV's action space to continuous action. This is discussed in detail in the second experiment (in Section 6.2).

This design would also help to understand the difference in the performance of the same RL algorithm, i.e. PPO with and without CBF as a safety layer when the algorithm is tested on the MARL-CAV environment. The expectation is that in second experiment the number collisions would decrease and agents reach optimal policy faster compared to first experiment.

In terms of the evaluation metrics, the second experiment should have a higher average evaluation reward and better average episode length compared to the first experiment

6.1 MARL-CAV validation

This experiment simulates lane changing in CAV using the PPO RL algorithm. This experiment aims to ensure that the implementation of MARL-CAV is valid where the agents can learn lane changing after a few training episodes. In this experiment, the vehicle aims to learn to safely change lanes avoiding collision with other CAVs or HDVs while taking discrete actions. The details of the set-up are as follows:

1. Number of CAVs - 2
2. Number of HDVs - 6
3. Action type - Discrete (left, right, forward, idle)
4. Reward Function
    - Penalty on collision

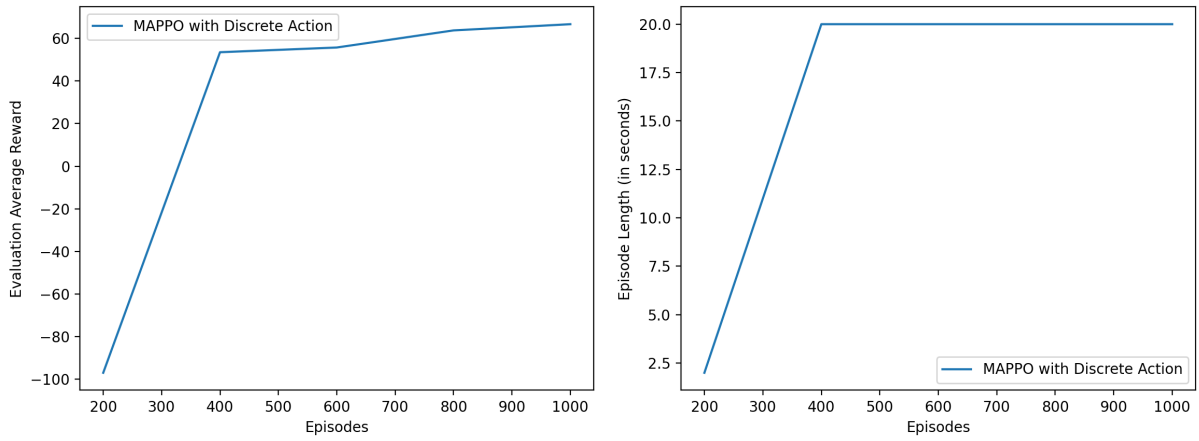
- Reward for headway

- Reward for lane change

- Reward for right lane

5. Number of training episodes - 1000

6. Number of Evaluation episodes - 3



(a) Average evaluation reward

(b) Average episode length

Figure 6.1: Metrics for MARL-CAV validation with discrete action space and 1000 training episodes

From Figure 6.1a we can observe that the average rewards of evaluation episodes have increased over the number of training episodes. In the beginning of training, the average reward was negative because there must be a lot of collisions resulting in higher penalties and negative rewards. The policy would not be good during the initial training episodes.

As the training reached 350 episodes, the policy improved, resulting in fewer or no collisions and higher rewards.

From Figure 6.1b we can observe that the average length of evaluation episodes has increased to the maximum episode length. Like in 6.1b, after around 350 training episodes, the episodes do not terminate from collisions and run till the maximum duration

From both the metric values and graphs, it can be inferred that the MARL-CAV set-up is valid, and the agents can learn to make lane changes safely.

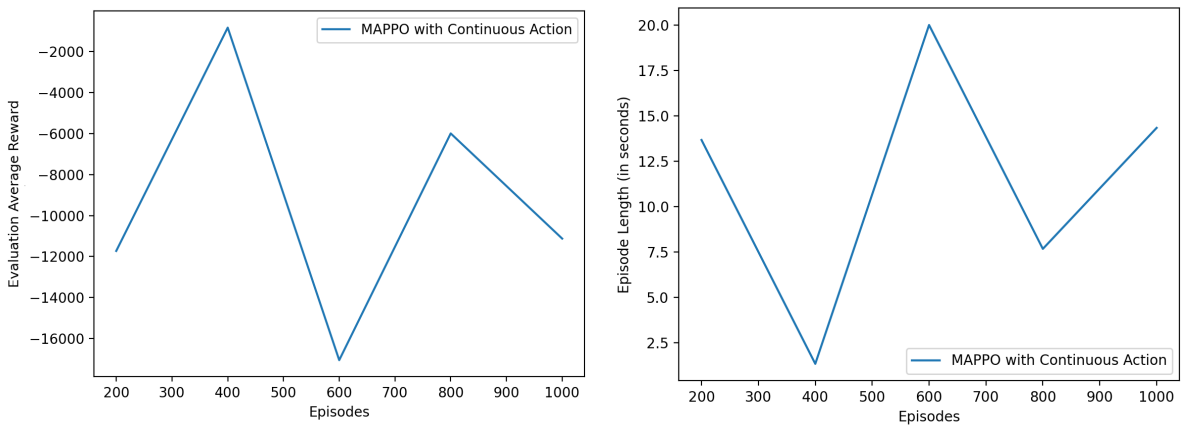
6.2 MARL-CAV with control barrier function

This experiment aims to simulate safe lane changing in CAVs using the control barrier functions with PPO RL algorithm. As discussed in Section 6.0.2, for I had to first modify the MARL-

CAV implementation to work with continuous action space. This is explained in detail in Section 5.2.3.

Before checking the performance of CBF with MARL-CAV, one important step is to validate the performance of PPO algorithm with continuous action space. In this experiment, the vehicle aims to learn to safely change lanes avoiding collision with other CAVs or HDVs while taking continuous actions. The details of the set-up are as follows:

1. Number of CAVs - 2
2. Number of HDVs - 6
3. Action type - Continuous (Steering angle and Throttle)
4. Reward Function
    - Penalty on collision
    - Reward for headway
    - Reward for lane change
    - Reward for right lane
    - Reward for vehicle to stay in lane
5. Number of training episodes - 1000
6. Number of Evaluation episodes - 3



(a) Average evaluation reward

(b) Average episode length

Figure 6.2: Metrics for MARL-CAV validation with continuous action space and 1000 training episodes

From Figure 6.2 we observe that MARL-CAV with continuous action space cannot achieve optimal policy. The rewards in 6.2a are high negative values indicating poor policy and a high

number of collisions. The zig-zag nature of the graph shows that the policy is unstable and the agent is not learning.

In Figure 6.2b we can see similar observation; the evaluation episodes terminate before reaching maximum episode length i.e. 20 seconds.

It is clear from the metrics that MARL-CAV is not able to achieve stable performance with continuous action.

As the MARL-CAV is not working properly with continuous action space I could not work further on it to integrate the control barrier functions as the control barrier functions only work with continuous action space.

6.3 Discussion

The results in the first experiment (in Section 6.1) validated the implementation of MARL-CAV. The values of evaluation metrics increased over time (in Figure 6.1), indicating agents are able to learn a stable policy.

However, the results of second experiment 6.2 are not good. The values of evaluation metrics oscillate over time (in Figure 6.2) and show high negative rewards for the entire duration of the training, this indicates that agents are unable to learn a stable policy when a continuous action space was used. As discussed in the implementation section 5.2.3 the continuous action did work for the OpenAI inverted pendulum environment (Brockman et al. 2016). As this experiment has failed, the integration of CBF with MARL-CAV is unsuccessful.

I spent a significant amount of time (over a week) debugging the cause of this poor performance of MARL-CAV with continuous action but could not fix the issue. While simulating this, I observed that the agents often leave the lanes. It could be because the RL policy does not control the steering angle and throttle to appropriate values. Due to this, I even added a reward to keep the agents at the centre of the lane, but still, there were no improvements. The issue might be with the source code of the simulation library highway-env (Leurent 2018). However, it is difficult to pinpoint the issue without more time and detailed analysis.

The RL algorithm could not learn an optimal policy in the second experiment. Hence, I could not apply the control barrier approach to MARL-CAV. Since the control barrier function approach only works with continuous action space, this poses a significant drawback in the application of CBFs, as the majority of the papers in the literature that have implemented MARL for lane changing in CAVs have used a discrete action space Chen et al. (2022), Zhou et al. (2021), Ye et al. (2020) compared to a few Fu et al. (2020) who have used continuous action space.

From the experiments, it can be inferred that the integration of existing safety approaches

like CBF to multi-agent RL in CAVs is not straight-forward and is a significant engineering challenge. Even though this experiment failed, I kept this to highlight the limitations of existing safety approach for MARL and the challenges in developing safe MARL for connected autonomous vehicles.

7 Conclusion

This Chapter summarises the work presented in this report, highlight the challenges encountered, discusses future work and concludes with a reflection of the author on his experience doing this project.

7.1 Summary

This dissertation developed a multi-agent reinforcement learning (MARL) based lane changing algorithm for connected autonomous vehicles (MARL-CAV). Parameter sharing and replay-buffer schemes were used to ensure collaboration among connected vehicles. The implementation of MARL-CAV was validated in the experiment in Section 6.1. The main intention of developing the MARL-CAV model was to examine whether external safety supervisors can help enhance the safety of MARL for connected autonomous vehicles in a mixed traffic scenario. As this topic is quite advanced, most of the papers reviewed in this dissertation were published in last three years. The investigation showed that most of the state-of-the-art implementations of MARL for lane changing in CAVs use a custom reward function to introduce safety, but this is not completely safe as the agents still visit unsafe states during the exploration phase of the RL algorithm.

State-of-the-art external safety techniques for MARL were extensively researched and analysed. It was observed that broadly these techniques come with a fair share of assumptions that can be classified into two groups; the first type of technique uses some prior information about unsafe states to develop safety critic which can provide the probability of agents being in unsafe situations in future states, and uses a separate recovery policy to bring the agent back to safe states. The model predictive control and Recovery RL safety approaches belong to this category. The second type of technique creates a set of safe states by estimating the dynamics of the system. They learn about unsafe states and project the actions of the reinforcement learning (RL) agent into the safe set and constrain unsafe action. The shielding and Control barrier functions safety approaches belong to this second category.

The characteristics of safe MARL for lane changing scenarios were discussed (summarised in Table 4.1), and it was inferred that an ideal safety supervisor approach should support

both continuous and discrete actions, should operate in a mixed traffic scenario, should not require a recovery policy and any prior information about unsafe states. Based on these, the functioning and limitations of different external safety supervisors for multi-agent RL were analysed and summarised in Table 4.2. It was observed that these safety techniques come with many assumptions like availability of recovery policies, prior information about unsafe states, the requirement of continuous action space etc. and are often not simulated in mixed agents (controlled and uncontrolled) scenarios.

These assumptions mean these techniques might not be directly applied to MARL for CAVs. To further verify the same, an experiment (in Section 6.2) was performed where the MARL-CAV setup was modified (from discrete actions to control action space) to support control barrier functions. However, the MARL-CAV implementation with continuous action could not reach a stable policy, and hence it could not be verified whether external safety technique control barrier functions can enhance the safety of MARL-CAV. However, the experiment highlights that integrating these safety approaches to MARL-CAV is not straightforward and requires significant work, which involves modifying either the MARL-CAV implementation or the safety techniques.

7.2 Challenges

The papers reviewed while studying literature for this dissertation were published quite recently. Most of them being published in the last three years. This made the project interesting but equally challenging.

During the state-of-the-art research, it was observed that there was plenty of research on different AI approaches for motion planning in CAVs. However, there is not much work showing research on safe AI approaches for motion planning in CAVs. This made it difficult to investigate the research questions. In addition, though some external safety approaches for MARL were found, these were never applied in a complex setting like lane changing for connected autonomous vehicles.

Moreover, the source code of these safety approaches is not open-sourced; hence, implementing these from scratch is challenging and requires a lot of time. Even if some of them are open-sourced, it is still a challenge to integrate these with a new task like MARL-CAV in this dissertation because the initial implementation is mostly done for a specific environment.

Apart from the above challenges, some were faced during the simulation environment development. These were discussed in detail in the section 5.2. Even after solving these challenges, modifying the simulation environment to work with continuous action space was still a huge task. This issue was unresolved due to the limited time.

Despite these challenges, significant work was done to solve these issues and answer the stated

research question.

7.3 Future Work

This dissertation aimed to contribute to the noteworthy research gap of safe MARL for CAVs and leaves plenty of room for future work. In this dissertation, only one (i.e., control barrier functions) out of the four (shielding, control barrier functions, model predictive control and Recovery RL) discussed safety approaches was experimented with. In future work, the suitability of the next most promising approach, as per the analysis table 4.2i.e., Recovery RL, can be examined.

The experiments in Sections 6.1 and 6.2 were done with the default number of CAVs and HDVs. This could be changed to increase the penetration rate of CAVs, and variance in results can be studied. Also, the maximum duration of episodes in the experiments was 20 seconds. This could be increased to verify those stable RL policies that perform well, even for longer simulations.

Apart from this, in this dissertation, the proximal policy optimization (PPO) algorithm is used. However, other policy-based RL algorithms can also be applied, and the performance of multiple algorithms can be compared.

7.4 Dissertation Reflection

Deep learning has increased interest in adding new systems to which artificial intelligence (AI) will be applied. Reinforcement learning (RL) has also had a renaissance, with Deep RL models created to compete with experts in games like Go and Atari.

Nevertheless, RL is still having trouble being adopted and successfully used in various applications, particularly regarding safety-critical jobs like robotics, autonomous driving, and industrial control. This dissertation was motivated by the idea that safe RL is an open problem.

Generally, in reinforcement learning papers, the reward function is modified to ensure agents' safety, but this does not completely ensure safety as the agents still visit unsafe states. This would be a problem when RL is used in real-life safety-critical applications, e.g. application of RL in lane changing for connected automated vehicles. Hence, the research question was to verify whether an external safety supervisor can be employed to enhance the safety of MARL for CAVs.

The research question was simple, and finding the answer was very challenging. In the process, I connected with some researchers and working professionals in top the self-driving industry

through email. It was great to see these people are open to helping and discuss the theoretical and practical challenges I faced while working on this dissertation. As discussed in the conclusion, the second experiment to study the use of control barrier functions (CBF) in MARL-CAV was unsuccessful. This motivated the question: Are these safety methods scalable and applicable to MARL-CAV environments? Initially, my research question was specific to CBFs, but when this did not work, I decided to pivot the research question a bit and with the limited time left, I decided to perform an extensive theoretical analysis of all four external safety approaches. Interestingly, the conclusion was similar, none of these approaches can be directly applied to MARL-CAV, and their integration into MARL-CAV is challenging. At the end of this dissertation, I can say that creating safe MARL for CAVs is a challenging task, and there is no easy option to solve it. It is still an open problem, and current research is far from building truly safe MARL approaches for connected autonomous vehicles.

I started with limited knowledge of reinforcement learning and connected autonomous vehicles. I am glad that this dissertation has helped me learn and understand reinforcement learning in more detail. This has also helped me to understand the challenges one can face when research ideas are applied to simulate real-world scenarios. Thanks to Trinity College Dublin and Prof. Melanie for giving this opportunity to work on state-of-the-art research problems.

Bibliography

Achiam, J. (2018), ‘Spinning Up in Deep Reinforcement Learning’.

Ames, A. D., Xu, X., Grizzle, J. W. & Tabuada, P. (2017), ‘Control barrier function based quadratic programs for safety critical systems’, *IEEE Transactions on Automatic Control* 62(8), 3861–3876.

Bae, S.-H., Joo, S.-H., Pyo, J.-W., Yoon, J.-S., Lee, K. & Kuc, T.-Y. (2020), Finite state machine based vehicle system for autonomous driving in urban environments, *in* ‘2020 20th International Conference on Control, Automation and Systems (ICCAS)’, pp. 1181–1186.

Berkenkamp, F., Turchetta, M., Schoellig, A. P. & Krause, A. (2017), Safe model-based reinforcement learning with stability guarantees, *in* ‘Proceedings of the 31st International Conference on Neural Information Processing Systems’, NIPS’17, Curran Associates Inc., Red Hook, NY, USA, p. 908–919.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. & Zaremba, W. (2016), ‘Openai gym’.

Cai, Z., Cao, H., Lu, W., Zhang, L. & Xiong, H. (2021), ‘Safe multi-agent reinforcement learning through decentralized multiple control barrier functions’.

Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M. & Spanò, S. (2021), ‘Multi-agent reinforcement learning: A review of challenges and applications’, *Applied Sciences* 11(11).
URL: *https://www.mdpi.com/2076-3417/11/11/4948*

Chen, D., Li, Z., Hajidavalloo, M., Chen, K., Wang, Y., Jiang, L. & Wang, Y. (2022), ‘Deep multi-agent reinforcement learning for highway on-ramp merging in mixed tra□c’.

Chen, Y., Lu, C. & Chu, W. (2020), ‘A cooperative driving strategy based on velocity prediction for connected vehicles with robust path-following control’, *IEEE Internet of Things Journal* 7(5), 3822–3832.

Chen, Y., Zha, J. & Wang, J. (2019), 'An autonomous t-intersection driving strategy considering oncoming vehicles based on connected vehicle technology', *IEEE/ASME Transactions on Mechatronics* 24(6), 2779–2790.

Cheng, R., Orosz, G., Murray, R. M. & Burdick, J. W. (2019), End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks, *in* 'Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence', AAAI'19/IAAI'19/EAAI'19, AAAI Press.
URL: *https://doi-org.elib.tcd.ie/10.1609/aaai.v33i01.33013387*

Choi, J., Castañeda, F., Tomlin, C. J. & Sreenath, K. (2020), 'Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions'.
URL: *https://arxiv.org/abs/2004.07584*

Chow, Y., Ghavamzadeh, M., Janson, L. & Pavone, M. (2017), 'Risk-constrained reinforcement learning with percentile risk criteria', 18(1), 6070–6120.

Chu, T., Chinchali, S. & Katti, S. (2020), 'Multi-agent reinforcement learning for networked system control'.
URL: *https://arxiv.org/abs/2004.01339*

Commission, E. (2022), 'Europe commission'.
URL: *shorturl.at/JOX03*

Darapaneni, N., R, P. R., Reddy Paduri, A., Anand, E., Rajarathinam, K., Eapen, P. T., K, S. & Krishnamurthy, S. (2021), Autonomous car driving using deep learning, *in* '2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)', pp. 29–33.

Dixit, V. V., Chand, S. & Nair, D. J. (2016), 'Autonomous vehicles: Disengagements, accidents and reaction times', *PLOS ONE* 11(12).

DOT, U. (2022), 'Fy 2022-26 u.s. dot strategic plan'.
URL: *https://www.transportation.gov/dot-strategic-plan*

Ehlers, R. & Raman, V. (2016), Slugs: Extensible gr(1) synthesis, *in* S. Chaudhuri & A. Farzan, eds, 'Computer Aided Verification', Springer International Publishing, Cham, pp. 333–339.

ElSayed-Aly, I., Bharadwaj, S., Amato, C., Ehlers, R., Topcu, U. & Feng, L. (2021), Safe multi-agent reinforcement learning via shielding, *in* 'Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems', AAMAS '21, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 483–491.

Emam, Y., Glotfelter, P., Kira, Z. & Egerstedt, M. (2021), 'Safe model-based reinforcement learning using robust control barrier functions'.
URL: *https://arxiv.org/abs/2110.05415*

Favarò, F. M., Nader, N., Eurich, S. O., Tripp, M. & Varadaraju, N. (2017), 'Examining accident reports involving autonomous vehicles in california', *PLOS ONE* 12(9).

Fisac, J. F., Bronstein, E., Stefansson, E., Sadigh, D., Sastry, S. S. & Dragan, A. D. (2019), Hierarchical game-theoretic planning for autonomous vehicles, *in* '2019 International Conference on Robotics and Automation (ICRA)', pp. 9590–9596.

Fu, Y., Li, C., Yu, F. R., Luan, T. H. & Zhang, Y. (2020), 'An autonomous lane-changing system with knowledge accumulation and transfer assisted by vehicular blockchain', *IEEE Internet of Things Journal* 7(11), 11123–11136.

García, J. & Fernández, F. (2015), 'A comprehensive survey on safe reinforcement learning', 16(1), 1437–1480.

Geibel, P. & Wysotzki, F. (2005), 'Risk-sensitive reinforcement learning applied to control under constraints', *J. Artif. Int. Res.* 24(1), 81–108.

Google (2022), 'Google's waymo'.
URL: *https://waymo.com/company/*

Grandia, R., Taylor, A. J., Ames, A. D. & Hutter, M. (2021), Multi-layered safety for legged robots via control barrier functions and model predictive control, *in* '2021 IEEE International Conference on Robotics and Automation (ICRA)', pp. 8352–8358.

Gupta, J. K., Egorov, M. & Kochenderfer, M. (2017), Cooperative multi-agent control using deep reinforcement learning, *in* G. Sukthankar & J. A. Rodriguez-Aguilar, eds, 'Autonomous Agents and Multiagent Systems', Springer International Publishing, Cham, pp. 66–83.

Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. (2018), Soft actor-critic: O□-policy maximum entropy deep reinforcement learning with a stochastic actor, *in* 'International conference on machine learning', PMLR, pp. 1861–1870.

Haydari, A. & Yılmaz, Y. (2022*a*), 'Deep reinforcement learning for intelligent transportation systems: A survey', *IEEE Transactions on Intelligent Transportation Systems* 23(1), 11–32.

Haydari, A. & Yılmaz, Y. (2022*b*), 'Deep reinforcement learning for intelligent transportation systems: A survey', *IEEE Transactions on Intelligent Transportation Systems* 23(1), 11–32.

Innovation, B. (2022), 'Benz innovation'.
URL: *https://www.mercedes-benz.com/en/innovation/autonomous/*

Ji, A. & Levinson, D. (2020), 'A review of game theory models of lane changing', *Transportmetrica A: Transport Science* 16(3), 1628–1647.

Kaushik, M., Singhania, N., S., P. & Krishna, K. M. (2019), Parameter sharing reinforcement learning architecture for multi agent driving, *in* 'Proceedings of the Advances in Robotics 2019', AIR 2019, Association for Computing Machinery, New York, NY, USA.
URL: *https://doi.org/10.1145/3352593.3352625*

Kesting, A., Treiber, M. & Helbing, D. (2010), 'Connectivity statistics of store-and-forward intervehicle communication', *IEEE Transactions on Intelligent Transportation Systems* 11(1), 172–181.

Khayatian, M., Mehrabian, M., Allamsetti, H., Liu, K.-W., Huang, P.-Y., Lin, C.-W. & Shrivastava, A. (2021), *Cooperative driving of connected autonomous vehicles using responsibility-sensitive safety (RSS) rules*, Association for Computing Machinery.

Koller, T., Berkenkamp, F., Turchetta, M. & Krause, A. (2018), Learning-based model predictive control for safe exploration, *in* '2018 IEEE conference on decision and control (CDC)', IEEE, pp. 6059–6066.

Kulkarni, R., Dhavalikar, S. & Bangar, S. (2018), Tra□c light detection and recognition for self driving cars using deep learning, *in* '2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)', pp. 1–4.

Kupferman, O. & Vardi, M. Y. (2001), 'Model checking of safety properties', *Formal Methods in System Design* 19(3), 291–314.
URL: *https://doi.org/10.1023/A:1011254632723*

Kurzhanski, A. B. & Varaiya, P. (2000), Ellipsoidal techniques for reachability analysis, *in* N. Lynch & B. H. Krogh, eds, 'Hybrid Systems: Computation and Control', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 202–214.

Le Cleac'h, S., Schwager, M. & Manchester, Z. (2021), 'ALGAMES: A Fast Augmented Lagrangian Solver for Constrained Dynamic Games', *Autonomous Robots* .
URL: *https://arxiv.org/pdf/2104.08452.pdf*

Leurent, E. (2018), 'An environment for autonomous driving decision-making', `https://github.com/eleurent/highway-env`.

Li, B., Li, Z., Yue, Z., Youmin, J. & Ning (2018), 'Cooperative lane change motion planning of connected and automated vehicles: A stepwise computational framework', *IEEE Intelligent Vehicles Symposium (IV)* pp. 334–338.

Li, J., Fu, D., Yuan, Q., Zhang, H., Chen, K., Yang, S. & Yang, F. (2019), 'A tra□c prediction enabled double rewarded value iteration network for route planning', *IEEE Transactions on Vehicular Technology* 68(5), 4170–4181.

Liao, X., Zhao, X., Wang, Z., Han, K., Tiwari, P., Barth, M. J. & Wu, G. (2021), 'Game theory-based ramp merging for mixed tra□c with unity-sumo co-simulation', *IEEE Transactions on Systems, Man, and Cybernetics: Systems* pp. 1–12.

Lin, D., Li, L. & Jabari, S. E. (2019), 'Pay to change lanes: A cooperative lane-changing strategy for connected/automated driving', *Transportation Research Part C: Emerging Technologies* 105(11), 550–564.

Littman, M. L. (1994), Markov games as a framework for multi-agent reinforcement learning, *in* 'Proceedings of the Eleventh International Conference on International Conference on Machine Learning', ICML'94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 157–163.

Liu, J. (2020), Survey of the image recognition based on deep learning network for autonomous driving car, *in* '2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)', pp. 1–6.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P. & Wießner, E. (2018), Microscopic tra□c simulation using sumo, *in* 'The 21st IEEE International Conference on Intelligent Transportation Systems', IEEE.
URL: *https://elib.dlr.de/124092/*

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P. & Mordatch, I. (2017), Multi-agent actor-critic for mixed cooperative-competitive environments, *in* 'Proceedings of the 31st International Conference on Neural Information Processing Systems', NIPS'17, Curran Associates Inc., Red Hook, NY, USA, p. 6382–6393.

Martens, M. & van den Beukel, A. (2013), The road to automated driving: Dual mode and human factors considerations, *in* '16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)', pp. 2262–2267.

Motors, G. (2022).
URL: *https://www.gm.com/stories/self-driving-cars*

Okuyama, T., Gonsalves, T. & Upadhay, J. (2018), Autonomous driving system based on deep q learnig, *in* '2018 International Conference on Intelligent Autonomous Systems (ICoIAS)', pp. 201–205.

Paret, D., Rebaine, H. & Engel, B. A. (2022), *The Buzz about Autonomous and Connected Vehicles*, Wiley, pp. 3–22.

Qin, Z., Zhang, K., Chen, Y., Chen, J. & Fan, C. (2021), 'Learning safe multi-agent control with decentralized neural barrier certificates'.

Rana, M. M. & Hossain, K. (2021), 'Connected and autonomous vehicles and infrastructures: A literature review', *International Journal of Pavement Research and Technology* .

Russell, S. J. & Norvig, P. (2016), 'Artificial intelligence: a modern approach. malaysia'.

SAE (2022), 'Sae levels of driving automation™ refined for clarity and international audience'.
URL: *https://www.sae.org/blog/sae-j3016-update*

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. (2017), 'Proximal policy optimization algorithms'.

Shladover, S. E. (2018), 'Connected and automated vehicle systems: Introduction and overview', *Journal of Intelligent Transportation Systems* 22(3), 190–200.

Takehara, R. & Gonsalves, T. (2021), Autonomous car parking system using deep reinforcement learning, *in* '2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)', pp. 85–89.

Tesla's Autopilot (2022).
URL: *https://tesla.com/autopilot*

Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K., Hwang, M., Gonzalez, J. E., Ibarz, J., Finn, C. & Goldberg, K. (2020), 'Recovery rl: Safe reinforcement learning with learned recovery zones'.
URL: *https://arxiv.org/abs/2010.15920*

USDOT (2022), 'United states. department of transportation. intelligent transportation systems (its) its strategic plan, 2015-2019.'.
URL: *https://rosap.ntl.bts.gov/view/dot/3506*

Veres, M. & Moussa, M. (2020), 'Deep learning for intelligent transportation systems: A survey of emerging trends', *IEEE Transactions on Intelligent Transportation Systems* 21(8), 3152–3168.

Wu, C., Kreidieh, A. R., Parvate, K., Vinitsky, E. & Bayen, A. M. (2022), 'Flow: A modular learning framework for mixed autonomy tra□c', *IEEE Transactions on Robotics* 38(2), 1270–1286.

Ye, F., Cheng, X., Wang, P., Chan, C.-Y. & Zhang, J. (2020), Automated lane change strategy using proximal policy optimization-based deep reinforcement learning, *in* '2020 IEEE Intelligent Vehicles Symposium (IV)', pp. 1746–1752.

Zanon, M. & Gros, S. (2021), 'Safe reinforcement learning using robust mpc', *IEEE Transactions on Automatic Control* 66(8), 3638–3652.

Zhang, K., Yang, Z. & Başar, T. (2019), 'Multi-agent reinforcement learning: A selective overview of theories and algorithms'.
URL: *https://arxiv.org/abs/1911.10635*

Zhang, W., Bastani, O. & Kumar, V. (2019), 'Mamps: Safe multi-agent reinforcement learning via model predictive shielding'.
URL: *https://arxiv.org/abs/1910.12639*

Zhao, H., Zeng, X., Chen, T., Liu, Z. & Woodcock, J. (2021), 'Learning safe neural network controllers with barrier certificates', *Formal Aspects of Computing* 33(3), 437–455.

Zheng, Y., Ran, B., Qu, X., Zhang, J. & Lin, Y. (2020), 'Cooperative lane changing strategies to improve tra□c operation and safety nearby freeway o□-ramps in a connected and automated vehicles environment', *IEEE Transactions on Intelligent Transportation Systems* 21(11), 4605–4614.

Zhou, W., Chen, D., Yan, J., Li, Z., Yin, H. & Ge, W. (2021), 'Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed tra□c', *arXiv* .

Zhu, C., Dastani, M. & Wang, S. (2022), 'A survey of multi-agent reinforcement learning with communication'.
URL: *https://arxiv.org/abs/2203.08975*